Concurrently Secure Compact Blind Signatures from Module-SIS

Olivier Blazy[1,2], Lola-Baie Mallordy[1,2], and Weiqiang Wen[3]

[1] LIX, École Polytechnique, Institut Polytechnique de Paris
[2] Inria
[3] Telecom Paris, Institut Polytechnique de Paris

Abstract. A blind signature scheme allows a user to interact with a signer to obtain a valid signature on a message, while ensuring that the signer cannot learn any information on the message being signed, nor link a given couple message-signature to the specific interaction that produced it (*blindness*). In round-optimal (*i.e.,* two-move) blind signature schemes, a user sends a request (typically a commitment) for a message, and the signer responds with a signature. To achieve blindness, the resulting blind signature usually consists of a zero-knowledge proof of knowledge of a valid signature from the signer on a request. This reliance on zero-knowledge proofs has become the main bottleneck in reducing the blind signature size. In particular, state-of-the-art lattice-based blind signature schemes are instantiated based on the zero-knowledge proof system from [Lyubashevsky et al., EUROCRYPT, 2022], which results in blind signatures of at least 22 KB.

In this work, we carefully design a blind signature protocol following the classical lattice-based Σ–protocol as in [Ducas et al., CRYPTO, 2013], so that the last component naturally forms a short preimage of the hash of the message, as a classical GPV signature. As a result, this removes the need for zero-knowledge proofs in the blind signature. Eventually, this design allows us to obtain a significantly more compact blind signature of size 4.7 KB, with concurrent security under the Module-SIS assumption. As a trade-off, our protocol may require more than one round with small probability, due to the rejection sampling in lattice-based Σ–protocols. Under our proposed parameters, however, the expected number of rounds for honest users can be as small as 1.1, which is very close to optimal. To minimize the number of rounds, we require users to prove that they failed to derive a blind signature in the previous round, before starting a new one. This technique yields a moderately looser bound on the expected number of rounds for malicious users, who will be forced to terminate in at most 2.6 rounds.

Keywords: Lattices · Blind Signature · Post-quantum cryptography

1 Introduction

Blind signatures are a fundamental cryptographic primitive that extend digital signatures with an additional property of *blindness*, ensuring that the signer

learns nothing about the message being signed. Originally introduced by Chaum in 1982 [13] for digital cash systems, blind signatures have since become a cornerstone tool for building privacy-preserving authentication mechanisms such as e-voting [23] and anonymous credentials [12]. They can be found in practical applications, such as Privacy Pass [15].

In a blind signature protocol, a user interacts with a signer to obtain a valid signature on a message without revealing the message's content. For instance, in an e-voting setting, a voter blinds their vote and submits it to the election authority, which verifies eligibility and signs the blinded vote. The voter then unblinds the signature and submits the signed vote anonymously to the ballot box. Even if the authority and ballot box later collude, the blindness property ensures that no link can be established between a voter and their vote. Early constructions required multiple rounds of interaction between the user and the signer [39,40,29], which limited their efficiency in practical deployments. A major breakthrough came with Fischlin's 2006 generic construction [21], achieving round optimality in the CRS model: the protocol only requires a single request from the user and a single response from the signer. This minimal interaction paved the way to practical and efficient blind signatures for real-world systems.

Lattice-based Blind Signatures. By adapting Fischlin's generic blind signature construction [21] to the lattice setting, the work of Agrawal et al. [1] achieves the most efficient lattice-based blind signature scheme with respect to the overall transcript size. In their design, the signer holds a GPV trapdoor [25] for a public matrix $\mathbf{A}$. A user initiates the protocol by sending a single commitment (its signature request)

$$h = \mathbf{B}\mathbf{x} + H(m) \mod q$$

to its message m for a random matrix $\mathbf{B}$ and a small error $\mathbf{x}$, hidden under the LWE assumption. The signer answers with a short preimage $\mathbf{A}\boldsymbol{\sigma} = h \mod q$ (the pre-signature) using its trapdoor. The blind signature consists of an encryption of $(\boldsymbol{\sigma}, \mathbf{x})$, a zero-knowledge proof (ZKP) of knowledge of a short witness $(\boldsymbol{\sigma}, \mathbf{x})$ satisfying

$$\mathbf{A}\boldsymbol{\sigma} - \mathbf{B}\mathbf{x} = H(m) \mod q,$$

and the well-formedness of the ciphertext. The ZKP is instantiated with a general-purpose lattice-based proof system such as [36]. They obtain small signature sizes of 44 KB, and especially a very short transcript of only 1.5 KB as the signature request consists of the single commitment h. However, the security of their scheme relies on the non-standard one-more-ISIS (OMISIS) assumption. Indeed, since the user does not have to prove that the signature request h was honestly computed, it can use the signer as a preimage oracle for any target of its choice.

The most compact lattice-based blind signature based on a *standard* assumption was proposed by Beullens et al. [7]. They follow a similar construction, but avoid the one-more-ISIS assumption by using a nested random oracle technique in the user's request

$$h = \mathbf{B}\mathbf{x} + H(m, G(\mathbf{x})) \mod q \tag{1}$$

such that the user cannot predict the value of h. They add a ZKP that h was honestly computed, to force the user to query both random oracles when computing a request h. Their blind signature also consists of a ZKP of knowledge of both the opening $\mathbf{x}$ and the signer's pre-signature. While this scheme achieves smaller signature sizes of about 22 KB (the shortest achieved so far), it incurs a larger transcript because of the ZKP of honest computation. Indeed, this requires proving hash preimages, which is a costly statement, even if the authors estimate that this can be proven in around 60 KB using the LaBRADOR proof system [8]. A recent work [9] actually provided benchmarks for this blind signature protocol, and reported a larger proof size of around 110 KB, but also a proving time of only 1.18s, which is way faster than the 10s originally estimated in [8] using Ligero++.

We observe that in both schemes, the blind signature size is largely dominated by the size of the zero-knowledge proofs. Within this framework, the efficiency of the current lattice-based ZKP systems is the main limitation to reducing the blind signature sizes. This naturally raises the following question:

Can we design shorter blind signatures by removing zero-knowledge proofs from the blind signature component?

We answer in the affirmative by constructing a scheme for which a blind signature only consists of a short preimage to a hash, and can be verified in the same way as a classical GPV signature.

1.1 Contributions

We propose a blind signature scheme based on the standard Module-SIS (MSIS) and NTRU assumptions. We prove its security in the concurrent setting *i.e.,* against an adversary starting several sessions in parallel, and not with a sequential order. We achieve short blind signature sizes of only 4.7 KB, which is more than four times smaller than the current shortest lattice-based blind signatures [7] of around 22 KB. On the downside, our protocol is no longer round optimal and has an average of $M = 1.1$ rounds. We prove that a looser bound of $\tilde{M} = 2.6$ rounds on average also holds for malicious users, who try to repeat more rounds than necessary. We enforce this by requiring users to prove that the previous round failed, before starting a new one. We stress that in most applications of blind signatures, such as e-voting or anonymous credentials, where one signer answers to many users that run only one or a few executions of the protocol, the signer's complexity is always the bottleneck. In our case, the signer's total number of rounds will converge to $M = 1.1$ times the number of honest users plus at most $\tilde{M} = 2.6$ times the number of malicious users.

1.2 Technical Overview

Hash-and-sign signature schemes can be broadly classified into two categories: *Full Domain Hash* (FDH) such as Falcon [22], or *Fiat-Shamir with aborts* (FS)

such as Dilithium or BLISS [17,16]. Both techniques offer their advantages: FDH schemes produce shorter signature with fast signing and verification times, while FS schemes provide smaller public keys and are easier to implement.

The standard approach to constructing blind signature schemes is to use both of these techniques back to back [7,1,42]: the user first requests an FDH signature from the signer, and the blind signature is then formed as a proof of knowledge of this FDH signature using a FS proof (*i.e.,* a Σ–protocol). As mentioned above, this is the blueprint followed by [1,7], which both use Falcon as the underlying FDH signature scheme. This method actually contains two distinct commitment phases: the signature request which is a commitment to the user's message, and the commitment phase of the Σ–protocol in the ZKP. To the best of our knowledge, no previous work tried to merge the two into a single protocol to avoid redundancy. We obtain compact blind signatures by embedding the commitment phase of the Σ–protocol into the user's signature request, which is then signed by the signer with an FDH scheme.

Our protocol. We design a blind signature as a short witness $\boldsymbol{\sigma}$ for the relation $\mathbf{A}\boldsymbol{\sigma} = H(m)$. We follow the framework of a classical lattice-based Σ–protocol. In such protocols, the commitment phase consists of choosing a gaussian mask $\mathbf{y}$, with a wide standard deviation, and sending a commitment $\mathbf{Ay}$ to the mask. We propose to use $\mathbf{Ay}$ in the signature request to hide the message, instead of $\mathbf{Bx}$ where $\mathbf{x}$ is a smaller error and $\mathbf{B}$ an independent matrix as in previous schemes (refer to Equation (1)). The idea of using the same matrix $\mathbf{A}$ as the signer's public key to blind $H(m)$ was already described in [30]. Their approach consisted of reusing the entropy of the mask $\mathbf{y}$, used to hide $H(m)$, to also hide the pre-signature $\boldsymbol{\sigma}\cdot$ However, the mask $\mathbf{y}$ that they use is smaller than $\boldsymbol{\sigma}$, so they decided to reveal only the lower bits of $\boldsymbol{\sigma}$, that were perfectly hidden by $\mathbf{y}$. They still rely on a ZKP to prove knowledge of the higher bits of $\boldsymbol{\sigma}$, even if they succeeded in considerably reducing the witness size compared to [1,7]. Here, we use a mask $\mathbf{y}$ that is significantly bigger than the pre-signature, more precisely big enough for rejection sampling, such that we can remove entirely the need for a ZKP. We keep the nested random oracle technique from Beullens et al. [7], such that our signature request has the following form:

$$h = (-1)^b\big(-\mathbf{Ay} + H(m,\rho)\big) \mod q, \tag{2}$$

with $\rho = G(b, \mathbf{y}, \mathbf{r})$, G and H two hash functions, a random bit b and a fresh nonce $\mathbf{r}$. The bit b will be used later by the user to perform bimodal rejection sampling [16], to ensure that its blind signature is independent of the pre-signature. The user computes a proof π of honest computation of h as in [7] to prove that it used both random oracles G and H. We note that, for this statement to be an NP statement, the oracles have to be instantiated with concrete hash functions. To this end, we perform our security proofs in two steps. First, we design an intermediate protocol that does not include any ZKP and prove its security in the random oracle model. Secondly, we instantiate this intermediate protocol with concrete hash functions, and prove that the security of our blind signature

reduces to the assumption that the instantiated protocol remains secure. This is the route taken in [7], where they reduce their security to the unforgeability of an underlying signature scheme. In our case, we cannot directly reduce our security to an underlying signature scheme, as we can no longer extract the signer's answer from our blind signature. This is why we need to rely on a different intermediate protocol, which is neither a blind signature nor a signature scheme.

Then, the user's signing request consists of the commitment h, the proof π, as well as encryptions-to-the-sky[4] of m and the randomness $(b, \mathbf{y}, \mathbf{r})$. The encryption-to-the-sky of m is used as a commitment by the signer to verify that the message used in each round remains the same throughout a single execution of the protocol. This ensures that, by repeating rounds in a single execution, an adversary can only obtain blind signature(s) for the same message. From a security point of view, this is crucial to achieve one-more unforgeability, which requires an adversary to forge $k+1$ blind signatures for $k+1$ different messages from only k executions of the protocol.

As in prior works, the signer responds with a pre-signature $\boldsymbol{\sigma}_0$, which is a short preimage of h. Then, by combining the relation $\mathbf{A}\boldsymbol{\sigma}_0 = h \mod q$ with Equation (2), we obtain the equality

$$\mathbf{A}(\mathbf{y} + (-1)^b \boldsymbol{\sigma}_0) = H(m, \rho) \mod q.$$

Thus, the user obtains a short preimage $\boldsymbol{\sigma} = \mathbf{y} + (-1)^b \boldsymbol{\sigma}_0$ of a hash of its message. We use bimodal rejection sampling to ensure that the distribution of $\boldsymbol{\sigma}$ is independent of the pre-signature $\boldsymbol{\sigma}_0$. Whenever a signature is rejected, the user restarts the protocol with the same message, but different randomness. It proves in zero-knowledge to the signer that the previous value of $\boldsymbol{\sigma}$ was indeed rejected, otherwise the signer does not send any more pre-signatures. We prove that this bounds the expected number of rounds, even for malicious users that might try to flood the signer. The bound we obtain for malicious users is looser than the one for honest users, since this is in the reverse setting of what is usually considered in rejection sampling: a mask $\mathbf{y}$ with a bounded norm, as enforced by a ZKP, but sampled from an arbritrary distribution, while the center $\boldsymbol{\sigma}_0$ follows a discrete gaussian.

Finally when $\boldsymbol{\sigma}$ passes the rejection, the user outputs the blind signature $(\boldsymbol{\sigma}, \rho)$ for m, where $\boldsymbol{\sigma}$ is a short witness for the relation $\mathbf{A}\boldsymbol{\sigma} = H(m, \rho) \mod q$. This corresponds to the final response in a classical Σ–protocol.

Blindness. The blindness of our protocol does not rely on the fact that *no information* on $\mathbf{y}$ is leaked, since the signer can easily try to guess it from the relation

$$\mathbf{y} = \boldsymbol{\sigma} - (-1)^b \boldsymbol{\sigma}_0. \tag{3}$$

It rather relies on the fact that each transcript is equally probable. Indeed, for all blind signatures $(\boldsymbol{\sigma}, \rho)$ on a message m, and for all transcripts $(h^*, \boldsymbol{\sigma}_0^*)$, by

[4] An encryption under a public key with no associated secret key.

combining Equations (2) and (3), we obtain that

$$\begin{aligned} h^* &= (-1)^b\big(- \mathbf{A}(\boldsymbol{\sigma} - (-1)^b\boldsymbol{\sigma}_0^*) + H(m,\rho)\big) \\ &= (-1)^b\big(- H(m,\rho) + (-1)^b h^* + H(m,\rho))\big) \\ &= h^* \mod q \end{aligned}$$

is always verified. Note that this always holds because ρ does not leak any information on the mask $\mathbf{y}$. This is why we need the nonce $\mathbf{r}$, to add entropy to $\rho = G(b, \mathbf{y}, \mathbf{r})$, otherwise an adversary could use it as a distinguisher to know which $\mathbf{y}$ was used, by simply guessing the bit b.

On the positive side, we eliminate the need for any independent ZKP in the blind signature, which allows us to achieve small blind signature sizes. On the other hand, due to use of rejection sampling, our protocol may require more than one round with a small probability. Moreover, while [7] requires only a modulus relatively larger than $\|\boldsymbol{\sigma}_0\|$, our setting requires a significantly larger one since it has to exceed $\|\boldsymbol{\sigma}\| = \|\mathbf{y} + (-1)^b\boldsymbol{\sigma}_0\|$. Concretely, they use a modulus of only 12 bits, while we have to use a modulus of 28 bits. These two drawbacks together lead to a slightly larger (expected) transcript size.

1.3 Related Work

Several lattice-based blind signature schemes [45,33,41,3,4,10] were found to have errors in their security proofs, making them susceptible to practical attacks [29]. While exposing the errors in these proofs, Hauck *et al.* [29] also proposed a solution based on the standard SIS assumption. However, their scheme is a 3-move construction achieving blindness in the weaker *honest signer model*, with an impractical signature size of more than 7 MB. The lattice-based blind signature scheme from [47] is based on rejection sampling. They obtain blind signatures of 35.4 KB, but only for a security level of 80 bits. Their expected repetition rate is quite high since they use rejection sampling at several stages of their protocol, and their base protocol is already not round optimal (2 complete rounds) without the rejection sampling.

The first round optimal scheme was proposed in [35], with a practical signature size around 150 KB. However, their signing running time is linear in the maximum number of signatures that can be produced, which requires the public key to be renewed regularly. The work of [1] avoids the limitation on the number of signatures per public key by adapting Fischlin's framework [21] to the lattice setting, but relies on the non-standard assumption one-more-ISIS. The work of [42] manages to avoid this non-standard assumption, and proposes a round-optimal scheme with slightly bigger signature sizes based on MSIS and MLWE. However, this blows up their communication cost, and they get a transcript size of around 850 KB. Finally, the recent work of [7] builds on the lattice adaptation of Fischlin's framework [1], and avoids the one-more-ISIS assumption by using a nested random oracle technique. They obtain shorter signatures, but a bigger transcript and overall communication cost. The work of [30] is also based

on standard assumptions, and obtains blind signature sizes slightly smaller than those in [1]. They get rid of the need of proving hash preimages, which results in a smaller transcript as in [7], and reduced the size of the ZKP witness in their blind signature. This work was later improved in [31] by introducing a new sampler, obtaining round optimality and shorter sizes. All of these works [35,1,42,7,30] compute their blind signature with an independent ZKP system. On the opposite side, BRaccoon [28] is a blind signature scheme relying on homomorphic encryption, that preserves the algebraic structure of the Raccoon signature scheme [43]. Their protocol is two rounds, and yields a heavy transcript size. We summarize all of these results in Table 1, and compare them against ours.

Schemes	**Rounds**	**Assumption**	**|Signature|**	**|Transcript|**
[35]	1	MSIS/MLWE	150 KB	16 MB
[42]	1	MSIS/MLWE/NTRU	100 KB	850 KB
[1]	1	OMISIS/NTRU	44 KB	1.5 KB
[7]	1	MSIS/MLWE/NTRU	22 KB	$\approx$ 110 KB
[30]	1.5	MSIS/MLWE	36.28 KB	53.21 KB
[31]	1	MSIS/MLWE	26.19 KB	36.15 KB
[28]	2	MSIS/MLWE	28 KB	847 KB
	E(Rounds)	**Assumption**	**|Signature|**	**E(|Transcript|)**
This work	≈ 1.1	MSIS/NTRU	4.70 KB	$\approx$ 120 KB

Table 1: Comparison of lattice-based blind signature schemes. We denote by OMISIS the one-more-ISIS assumption. Since the number of rounds of our protocol depends on the rejection sampling, we gave the expected number of rounds **E(Rounds)**, and the expected transcript size **E(|Transcript|)**. The number of rounds for [30] is 1.5 since it is one-round, plus a message from the signer. We note that all round-optimal schemes automatically are concurrently secure. We updated the sizes given in [7] with the new estimations from [9].

Other post-quantum schemes. A recent work [11] proposes a generic framework to turn any Hash-and-Sign signature scheme into a blind signature scheme. By instantiating their framework with multivariate and code-based assumptions (UOV and Wave), they obtain very short blind signature sizes from only 3.8 KB to 11 KB, using MPC-in-the-Head techniques. However, their signature request includes a ZKP that they instantiate with hash-based SNARKs, which yields a large transcript size of at least 300 KB. Another work [26] also proposes new frameworks to build 3-pass blind signature schemes from cryptographic group actions. The authors considered instantiations from isogeny-based and code-based cryptography, and obtained blind signature sizes from 4.5 to 64.7 KB respectively. However, while they obtain very small sizes when using isogenies, producing a single blind signature takes at least 4.7 s[5]. In a follow-up work [27],

[5] As mentioned above our signing protocol is also costly, around 1.2s, but this comes mainly from the user-side due to the ZKP computation while the signer's workload is cheap. In [26], the workload is approximately the same for the user and the signer.

they manage to reduce the signature size of their code-based solution to only 8.84 KB. Finally, the work of [6] obtains short blind signatures of only 6.7 KB by relying on the VOLE-in-the-head ZKP system, and the non-standard assumption of one-more preimage resistance of the MAYO signature scheme. Their scheme yields a short transcript of only 0.45 KB.

Outline. We provide some cryptographic background in Section 2. In Section 3 we describe our blind signature protocol, and prove its security in Section 4. Finally, in Section 5, we propose a parameter selection for our scheme.

2 Preliminaries

Notations. Sampling an element x uniformly from a set S is denoted by $x \leftarrow\!\!\$\ S$. By $\mathcal{A}$ and $\mathcal{C}$, we denote respectively the adversary and the challenger of a security experiment, which are probabilistic polynomial-time algorithms. By $[n]$, we denote the set $[1, \ldots, n]$. We use bold uppercase for matrices (*e.g.,* $\mathbf{A}$) and bold lowercase for vectors (*e.g.,* $\mathbf{x}$), while scalars are in plain. We denote by $P\langle A(x), B(y)\rangle$ a protocol P between two parties A and B with inputs x and y.

2.1 Blind Signatures

Blind signatures are a special kind of digital signatures with an additional property called *blindness*, on top of an adjusted notion of *unforgeability* [44]. In such schemes, a user interacts with a signer to obtain a signature on its message, with the guarantee that the signer should not be able to link a message-signature pair to a specific execution of the protocol (blindness). Additionally, we can no longer use the classical unforgeability notion for digital signatures, in which the challenger sees the signed messages in the signature queries, and the adversary is only asked to forge one signature on a new message that was not in the queries. Indeed, in blind signatures, the challenger cannot see the message inside a signing request. To account for this, *one-more unforgeability* is introduced: the adversary must produce more valid blind signatures than the number of signing queries.

In most applications of blind signatures, the signer can be required to handle multiple signing sessions in parallel. Thus, we consider security in the *concurrent* model, where the signing queries are not done sequentially but in parallel.

Definition 1 (Blind Signature). *A blind signature scheme* BS *is defined by the following algorithms:*

- $\mathsf{BS.KeyGen}(\lambda) \to (\mathsf{vk}, \mathsf{sk})$*: takes as input the security parameter* λ *and outputs a signing key* sk *and a verification key* vk*.*

This can become an issue in contexts such as e-voting, where the signer has to issue many signatures.

– $\mathsf{BS.Sign}\langle \mathsf{Signer}(\mathsf{vk}, \mathsf{sk}), \mathsf{User}(\mathsf{vk}, m \in \{0,1\}^*)\rangle \to (\bot, \mathsf{sig})$*: an interactive protocol between a* Signer *and a* User*, where the* User *takes as input a message* m *and the* Signer *a signing key* sk*. At the end of the protocol, the* User *obtains a signature* sig *on* m*, while the* Signer *has no outputs.*
– $\mathsf{BS.Verif}(\mathsf{vk}, m, \mathsf{sig}) \to b \in \{0,1\}$*: either accepts or rejects the signature* sig *on* m *with the verification key* vk*.*

Definition 2 (Correctness). *A blind signature scheme is correct if there exists a negligible* ε *such that for any message* $m \in \{0,1\}^*$*,*

$$\Pr\left[\mathsf{Verif}(\mathsf{vk}, m, \mathsf{sig}) = \top \;\middle|\; \begin{array}{l} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\lambda) \\ \mathsf{sig} \leftarrow \mathsf{BS.Sign}\langle \mathsf{Signer}(\mathsf{vk}, \mathsf{sk}), \mathsf{User}(\mathsf{vk}, m)\rangle \end{array}\right] \geq 1 - \varepsilon.$$

Definition 3 (One-more unforgeability). *A blind signature scheme is said to be one-more unforgeable if for any polynomial* $k \in \mathbb{N}^*$*, any adversary* $\mathcal{A}$ *has a negligible advantage in the game* $\mathsf{Exp}^{\mathsf{UNF}}(k, \mathcal{A})$ *as in Figure 1 i.e., there exists a negligible* ε *such that*

$$\mathsf{Adv}^{\mathsf{UNF}}(\mathcal{A}, k) := \Pr[1 \leftarrow \mathsf{Exp}^{\mathsf{UNF}}(\mathcal{A})] \leq \varepsilon.$$

$\mathsf{Exp}^{\mathsf{UNF}}(k, \mathcal{A})$:

1 : $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
2 : $\mathcal{A}$ runs concurrently k instances $\mathsf{BS.Sign}\langle \mathsf{Signer}(\mathsf{vk}, \mathsf{sk}), \mathcal{A}(\mathsf{vk})\rangle$
3 : $\{m_i, \mathsf{sig}_i\}_{i \in [k+1]} \leftarrow \mathcal{A}(\mathsf{vk})$
4 : **return** $(\forall i, j \in [k+1], m_i \neq m_j) \wedge \forall i \in [k+1], \mathsf{Verif}(\mathsf{vk}, m_i, \mathsf{sig}_i) = \top$

$\mathsf{Exp}^{\mathsf{BLIND}}(\mathcal{A})$:

1 : $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
2 : $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{sk})$
3 : $b \leftarrow\!\!\$\ \{0,1\}$
4 : $\mathcal{A}$ runs concurrently the two instances $\mathsf{sig}_b \leftarrow \mathsf{BS.Sign}\langle \mathcal{A}, \mathsf{User}(\mathsf{vk}, m_b)\rangle$ $\mathsf{sig}_{1-b} \leftarrow \mathsf{BS.Sign}\langle \mathcal{A}, \mathsf{User}(\mathsf{vk}, m_{1-b})\rangle$
5 : **if** $\mathsf{sig}_0 = \bot \vee \mathsf{sig}_1 = \bot$:
6 : **return** b
7 : $b' \leftarrow \mathcal{A}(\mathsf{sig}_0, \mathsf{sig}_1)$
8 : **return** $b = b'$

Fig. 1: The one-more unforgeability $\mathsf{Exp}^{\mathsf{UNF}}(k, \mathcal{A})$ and blindness $\mathsf{Exp}^{\mathsf{BLIND}}(\mathcal{A})$ experiments.

Definition 4 (Blindness). *A blind signature scheme is said to be blind if any adversary* $\mathcal{A}$ *has a negligible advantage in the game* $\mathsf{Exp}^{\mathsf{BLIND}}(\mathcal{A})$ *as in Figure 1 i.e., there exists a negligible* ε *such that*

$$\mathsf{Adv}^{\mathsf{BLIND}}(\mathcal{A}) := |\Pr[1 \leftarrow \mathsf{Exp}^{\mathsf{BLIND}}(\mathcal{A})] - 1/2| \leq \varepsilon.$$

Full-fledged blindness lets the adversary compute itself the keys, and gives vk to the challenger.

2.2 Proof of Knowledge

Let $R = \{(y, w)\} \in \{0,1\}^* \times \{0,1\}^*$ be an NP relation. A proof of knowledge for the relation R is a two-party protocol $\langle P, V \rangle$ between a prover P and a verifier V with soundness error ε if the two conditions hold:

- **Completeness:** if $(y, w) \in R$ then we have $\Pr[\langle P(y, w), V(y) \rangle = 1] = 1$.
- **Soundness:** for any PPT P, if $(y, w) \notin R$ then we have

$$\mathsf{Adv}_{\mathsf{SND}}(\mathcal{A}) := \Pr[\langle P(y, w), V(y) \rangle = 1] \leq \varepsilon.$$

A proof of knowledge is called zero-knowledge if for any $V(y)$, there exists a PPT simulator $S(y)$ producing a simulated transcript that is indistinguishable from the one of the real interaction between $P(y, w)$ and $V(y)$.

Definition 5 (Zero-Knowledge). *We say that a proof of knowledge for the relation R is zero-knowledge if there exists a simulator $\mathcal{S}$ such that, for any $(y, w) \in R$, any adversary $\mathcal{A}$ and negligible ε, we have*

$$\mathsf{Adv}_{\mathsf{ZK}}(\mathcal{A}) := |\Pr[1 \leftarrow \mathcal{A}(\langle P(y, w), V(y) \rangle)] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{S}(y))]| \leq \varepsilon.$$

We work with zero-knowledge proofs in their non-interactive form, by using the Fiat-Shamir transform.

2.3 Lattices and Discrete Gaussians

Let ϕ be the Euler totient function and Φ_m be the m-th cyclotomic polynomial. For $n = \phi(m)$ and an odd modulus q, denote $\mathbb{R}$ and $\mathbb{R}_q$ respectively to be the n-dimensional rings

$$\mathbb{R} = \mathbb{Z}[X]/\Phi_m(X), \ \mathbb{R}_q = \mathbb{Z}_q[X]/\Phi_m(X).$$

Let $s > 0$ and $\mathbf{v} \in \mathbb{R}^n$, the gaussian function $\rho_{s,\mathbf{v}}$ over $\mathbb{R}^n$ is defined by

$$\rho_{s,\mathbf{v}}(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{v}\|^2}{2s^2}\right).$$

In this work, we consider the discrete gaussian distribution over R^k centered at $\mathbf{v}$ with standard deviation s given by the distribution function

$$D^k_{s,\mathbf{v}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{v}}(\mathbf{x})}{\rho_s(R^k)},$$

where we omit the subscript $\mathbf{v}$ when it is zero.

An important notion related to discrete gaussians over lattices is the smoothing parameter [38].

Definition 6 (Smoothing parameter). *For any positive real $\varepsilon > 0$ and lattice Λ, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is defined as the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \varepsilon$.*

We use the following lemma from [38], which ties the smoothing parameter of a lattice Λ to the statistical closeness of a gaussian distribution to the uniform distribution modulo Λ. For two distributions P and Q, we denote their statistical distance by

$$\Delta(P,Q) = \frac{1}{2}\sum_{v} |\Pr[P=v] - \Pr[Q=v]| \leq 1.$$

Lemma 1 (Lemma 4.1 [38]). *For any $\varepsilon > 0$, $\mathbf{c} \in \mathbb{R}^n$ and lattice $\Lambda(\mathbf{B})$, for $s > \eta_\varepsilon(\mathbf{B})$, the statistical distance between $D_{s,\mathbf{c}} \bmod P(\mathbf{B})$ and the uniform distribution over $P(\mathbf{B})$ is at most*

$$\Delta\big(D_{s,\mathbf{c}} \bmod P(\mathbf{B}), U(P(\mathbf{B}))\big) \leq \varepsilon/2.$$

2.4 Lattice Trapdoors

We use the following algorithms to generate a random NTRU trapdoor, and to sample gaussian vectors in a given lattice coset such as in [22,19].

Lemma 2. *Let q be a prime. There exists the following PPT algorithms:*

- $\mathsf{TrapGen}(q)$: *outputs a pair $(\mathbf{A}, \mathbf{T})$ such that $\mathbf{A} \in R_q^{1\times 2}$ is indistinguishable from uniform under the NTRU assumption, and $\mathbf{T}$ is a short basis for the lattice $\Lambda_q^\perp(\mathbf{A})$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{t}, s; r)$: *takes as input the above pair $(\mathbf{A}, \mathbf{T})$, and target vector $\mathbf{t} \in R_q^n$ and a sufficiently large $s \geq \eta_\varepsilon(\mathbb{Z}^{2n})\|\tilde{\mathbf{T}}\|$ for $\|\tilde{\mathbf{T}}\|$ the Gram-Schmidt norm of the trapdoor, and outputs a vector $\mathbf{v}$ from a distribution within negligible distance to $D_{\Lambda_q^{\mathbf{t}}(\mathbf{A}),s}$ over the randomness of the seed r. For a fixed seed r, the algorithm is deterministic.*

We use pseudorandom function families to deterministically generate the randomness seed r used in SamplePre.

Definition 7 (Pseudorandom function family [46]). *Let $\mathcal{F} = \{F_s\}_{s\in S}$ be a family of keyed functions mapping D to R. The family $\mathcal{F}$ is pseudorandom if the PRF-advantage of any adversary $\mathcal{A}$ is negligible:*

$$\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{A}) := \big|\Pr[s \leftarrow\$ S \mid 1 \leftarrow \mathcal{A}^{F_s}] - \Pr[f \leftarrow\$ \Gamma_{D,R} \mid 1 \leftarrow \mathcal{A}^{f}]\big|.$$

2.5 Rejection Sampling

We use the bimodal rejection sampling from [16], due to its low rejection rate. Their result can be expressed as follow.

$\mathsf{Reject}(\mathbf{z}, \mathbf{v}, s)$:

1 : $u \leftarrow\$ [0,1)$

2 : **if** $u > \dfrac{1}{M \exp\left(-\frac{\|\mathbf{v}\|^2}{2s^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{v} \rangle}{s^2}\right)}$

3 : **return** $\perp$

4 : **else return** $(\mathbf{z}, \mathbf{v})$

Fig. 2: Bimodal rejection sampling algorithm from Ducas et al. [16]. We denote by $\mathsf{Reject}(\mathbf{z}, \mathbf{v}, s; u)$ the deterministic algorithm with input randomness u.

Lemma 3 (Bimodal rejection sampling [16]). *Let $V \subseteq R^k$ be a set of polynomials of norm at most T and $\rho : V \to [0,1]$ be a probability distribution. Fix the standard deviation $s = \gamma T$, where $\gamma > 0$. Let $M = \exp(1/(2\gamma^2))$. We consider the following algorithm:*

- *Sample $\mathbf{v} \leftarrow\$ \rho$, $b \leftarrow\$ \{0,1\}$ and $\mathbf{y} \leftarrow\$ D_s^k$, set $\mathbf{z} = \mathbf{y} + (-1)^b \mathbf{v}$, and return $(\mathbf{z}, \mathbf{v}) \leftarrow \mathsf{Reject}(\mathbf{z}, \mathbf{v}, s)$ as defined in Figure 2.*

Then, the probability that $(\mathbf{z}, \mathbf{v}) \neq \perp$ is at least $1/M$, and the distribution of $(\mathbf{z}, \mathbf{v})$ conditioned on the event that $\mathsf{Reject}(\mathbf{z}, \mathbf{v}, s) \neq \perp$ is identical to the product distribution $\rho \times D_s^k$.

2.6 Hardness Assumption

The security of our blind signature schemes relies on the well-known computational lattice problem Module-SIS [32], defined as follow.

Definition 8 (MSIS). *Given $\mathbf{A} \in R_q^{k \times m}$, the MSIS_B problem with parameters $k, m > 0$ and $0 < B$ asks to find $\mathbf{z} \in R^m$ such that $\mathbf{A}\mathbf{z} = 0 \mod q$ and $0 < \|\mathbf{z}\| \leq B$. The advantage of an adversary $\mathcal{A}$ in solving MSIS_B is*

$$\mathsf{Adv}_{\mathsf{MSIS}_B}(\mathcal{A}) := \Pr\left[0 < \|\mathbf{z}\| \leq B, \mathbf{A}\mathbf{z} = 0 \bmod q \mid \mathbf{A} \leftarrow R_q^{k \times m}, \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A})\right].$$

3 Blind Signature from MSIS

We describe our lattice-based blind signature scheme BS.

3.1 Our Blind Signature Protocol

The detailed key generation and verification algorithms of our blind signature protocol, as well as its subroutines, are given in Figure 3, and the signing protocol is given in Figure 4. Similarly to previous lattice-based blind signature schemes [1,7], our scheme is based on a GPV trapdoor. More precisely, we use an NTRU lattice $\Lambda_q^{\perp}(\mathbf{A})$ where $\mathbf{A} = (h, 1) \in R_q^{1 \times 2}$.

$\mathsf{BS.KeyGen}(\lambda)$:

$\kappa \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$(\mathbf{T}, \mathbf{A}) \leftarrow \mathsf{TrapGen}(q)$
$(\mathsf{pk}_\mathsf{E}, \mathsf{sk}_\mathsf{E}) \leftarrow \mathsf{E.KeyGen}(\lambda)$
$\mathsf{vk} = (\mathbf{A}, \mathsf{pk}_\mathsf{E}), \mathsf{sk} = (\mathbf{T}, \kappa)$
return $(\mathsf{vk}, \mathsf{sk})$

$\mathsf{BS.GenRq}(\mathsf{vk}, m, c_m, r_m, \mathsf{pnc}, \mathsf{prev}_\mathsf{U})$:

$b \leftarrow\!\!\$\ \{0,1\}$
$\mathbf{y} \leftarrow\!\!\$\ D_s^2$
$\mathbf{r} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$c_\mathsf{nc} \leftarrow F(\mathsf{nc})$
$\rho = G(b, \mathbf{y}, \mathbf{r})$
$h = (-1)^b(H(m,\rho) - \mathbf{A}\mathbf{y})$
$c \leftarrow \mathsf{E.Enc}\big(\mathsf{pk}_\mathsf{E}, (b, \mathbf{y}, \mathbf{r}); r\big)$
$\pi \leftarrow \Pi.\mathsf{Prove}(m, r, r_m, b, \mathbf{y}, \mathbf{r}, \mathsf{prev}_\mathsf{U})$
if first round:
  $\mathsf{rq} \leftarrow (h, c, \pi, c_\mathsf{nc}, c_m)$
else $\mathsf{rq} \leftarrow (h, c, \pi, c_\mathsf{nc}, \mathsf{pnc})$
return $(\mathsf{rq}, b, \mathbf{y}, \mathbf{r}, r, \mathsf{nc})$

$\mathsf{BS.Verif}(\mathsf{vk}, m, (\boldsymbol{\sigma}, \rho))$:

if $\big(\mathbf{A}\boldsymbol{\sigma} = H(m,\rho) \mod q\big)$
  $\wedge\ (\|\boldsymbol{\sigma}\| \le \beta)$:
  return $\top$
return $\bot$

$\mathsf{BS.GenPSig}(\mathsf{vk}, (\mathbf{T}, \kappa), \mathsf{rq}, c_\mathsf{pnc}, \mathsf{prev}_\mathsf{S})$:

if first round:
  Parse rq as $(h, c, \pi, c_\mathsf{nc}, c_m)$
else parse rq as $(h, c, \pi, c_\mathsf{nc}, \mathsf{pnc})$
  if $F(\mathsf{pnc}) \neq c_\mathsf{pnc}$:
    return $\bot$
if $\neg\Pi.\mathsf{Verif}(\mathsf{vk}, c_m, c, h, \mathsf{pnc}, \mathsf{prev}_\mathsf{S}, \pi)$:
  return $\bot$
$\mathsf{count} \leftarrow 0$
$k \leftarrow F_\kappa(h||\mathsf{count})$
$\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
while $\|\boldsymbol{\sigma}_0\| > \beta_0$:
  $\mathsf{count} \leftarrow \mathsf{count} + 1$
  $k \leftarrow F_\kappa(h||\mathsf{count})$
  $\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
$\mathsf{nc}_\mathsf{S} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
return $\mathsf{psig} = (\boldsymbol{\sigma}_0, \mathsf{nc}_\mathsf{S})$

Fig. 3: The KeyGen and Verif algorithms for our blind signature protocol BS, and the user GenRq and signer GenPSig subroutines used in the signing protocol. The hash functions $H : \{0,1\}^* \longrightarrow R_q$, $G : R_q^2 \to \{0,1\}^{2\lambda}$ and $F : \{0,1\}^* \longrightarrow \{0,1\}^{2\lambda}$ are modeled as random oracles. Let $\mathcal{F} = \{F_s\}_{s\in\{0,1\}^{2\lambda}}$ be a PRF family, E be any IND-CPA encryption scheme and Π be the non-interactive ZKP system defined in Section 3.4.

We equip the signer with a trapdoor for $\mathbf{A}$, that lets it sample relatively short discrete gaussian vectors over any lattice coset $\Lambda_q^t(\mathbf{A})$, as described in Lemma 2. We assume SamplePre is deterministic *i.e.,* always provides the same preimage when given the same input target. More precisely, the randomness of SamplePre is generated by a PRF, which takes a key κ part of the signer's secret key as seed, and the signing request as input.

At the beginning of the protocol, the user computes an encryption-to-the-sky c_m (an encryption under a public key without any associated secret key) of its message m. This value is only computed once in the first round, and will be kept

the same throughout the rest of the protocol. It is used both as a commitment to ensure that the user keeps the same message throughout a single execution of the protocol, and for efficiency such that our reduction does not need to call the extractor of the ZKP when simulating a signer.

We denote by GenRq the user's subroutine that computes a signature request (Figure 3). Similarly to the commitment phase in a Σ–protocol, the user samples a gaussian mask $\mathbf{y}$ with a wide standard deviation above the smoothing parameter of the lattice. It also samples a bit b for the bimodal rejection sampling. Next, using the nested random oracle technique from [7], the user hashes its randomness as $\rho = G(b, \mathbf{y}, \mathbf{r})$ to derive the hash of its message $H(m, \rho)$, where $\mathbf{r} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$ is a fresh nonce. It commits to the hash of its message as

$$h = (-1)^b(H(m,\rho) - \mathbf{A}\mathbf{y}) \mod q. \tag{4}$$

The user computes a ZKP to prove the honest computation of its signature request, such that it is forced to go through all the random oracles. We add another encryption-to-the-sky of the randomness $(b, \mathbf{y}, \mathbf{r})$ used in the request, such that our reduction does not need to call the extractor of the ZKP when simulating a signer. The user only needs to send c_m in the first round.

We denote by $\mathsf{GenPSig}$ the signer's subroutine that computes a pre-signature (Figure 3). The signer verifies the ZKP to ensure the well-formedness of the request, and samples $\boldsymbol{\sigma}_0$ using its trapdoor such that $\mathbf{A}\boldsymbol{\sigma}_0 = h \mod q$. Note that, if the user queries several times on the same h, it will always obtain the same preimage, as we use a deterministic trapdoor sampling.

Finally, the user performs bimodal rejection sampling as in Lemma 5 on

$$\boldsymbol{\sigma} = \mathbf{y} + (-1)^b\boldsymbol{\sigma}_0, \tag{5}$$

to make it independent of the signer's preimage $\boldsymbol{\sigma}_0$. The randomness used in the rejection is derived from two nonces, one chosen by the user and one by the signer. Whenever $\boldsymbol{\sigma}$ is rejected, the user restarts the protocol with same message and fresh randomness. In its new request to the signer, the user proves that the previous signature was rejected (see Section 3.2 for more details), in addition to the validity of its new request.

Finally, when $\boldsymbol{\sigma}$ is accepted by the rejection sampling, the user outputs the couple $(\boldsymbol{\sigma}, \rho)$ as its blind signature on m, whih can be verified by checking that $\boldsymbol{\sigma}$ is a short vector and $\mathbf{A}\boldsymbol{\sigma} = H(m,\rho) \mod q$.

Full-fledged blindness. We prove the weaker version of blindness, where the keys are not generated by the adversary, but we explain how to attain full-fledged blindness by modifying the setup. The signer sends a proof along with the public key $\mathbf{A}$, that $\mathbf{A}$ defines a lattice with a sufficiently small smoothing parameter. This essentially requires proving knowledge of a sufficiently short basis $\mathbf{T}$ of the lattice, which we describe in further details in Appendix B. As for the encryption-to-the-sky, if E is a standard Regev encryption, its public key $\mathsf{pk}_\mathsf{E} = (\mathbf{A}_\mathsf{E}, \mathbf{b}_\mathsf{E})$ can be sampled uniformly without a secret key. Thus, we can set $(\mathbf{A}_\mathsf{E}, \mathbf{b}_\mathsf{E}) = H(\mathsf{seed})$ for a random oracle H, to prevent the adversary from keeping any decryption

key. In the unforgeability proof, the challenger can obtain a valid decryption key $\mathbf{s}$ by programming $H(\mathsf{seed}) = (\mathbf{A}_\mathsf{E}, \mathbf{b}_\mathsf{E} = \mathbf{A}_\mathsf{E}\mathbf{s} + \mathbf{e})$, which the adversary cannot distinguish from uniform keys under MLWE.

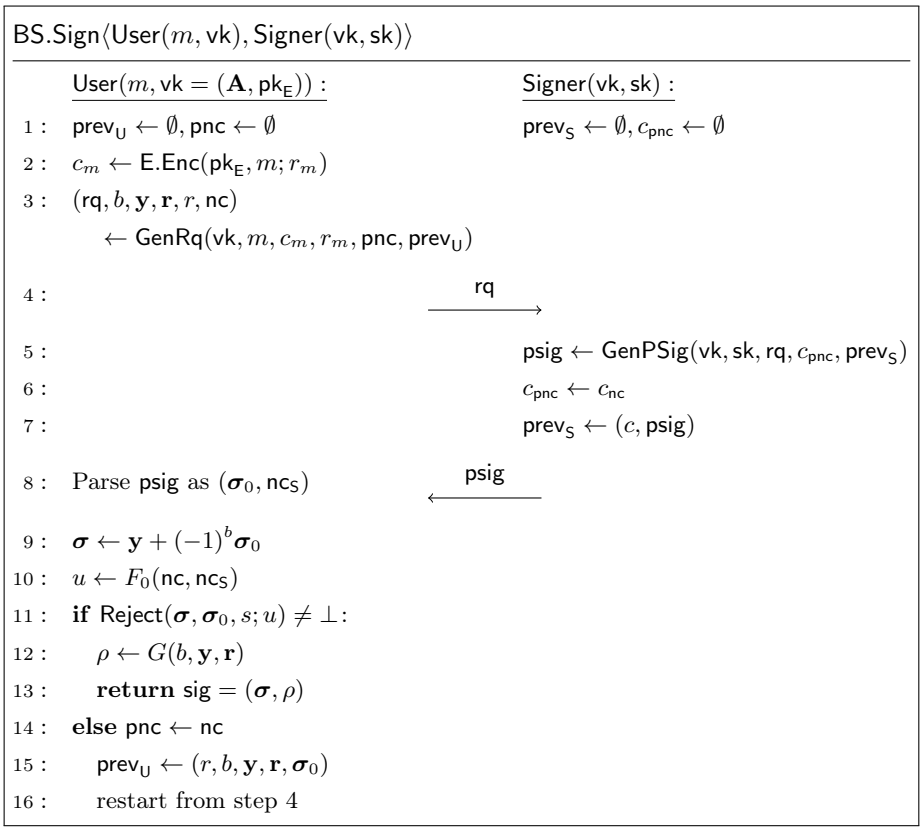

Fig. 4: The signing protocol of our blind signature scheme BS. Let $F_0 : \{0,1\}^* \longrightarrow [0,1)$ be a hash function modeled as a random oracle.

3.2 Proving Request Validity

We model the validity of a signing request as the relation R^{rq} for state st and witness w, defined as follow

$$R^{\mathsf{rq}} = \left\{ \begin{array}{l|l} & \rho = G(b, \mathbf{y}, \mathbf{r}) \\ & h_m = H(m, \rho) \\ \mathsf{st} = (\mathbf{A}, \mathsf{pk}_\mathsf{E}, c_m, c, h), & h = (-1)^b(h_m - \mathbf{A}\mathbf{y}) \mod q \\ w = (m, r, r_m, b, \mathbf{y}, \mathbf{r}) & c_m \leftarrow \mathsf{E.Enc}(\mathsf{pk}_\mathsf{E}, m; r_m) \\ & c \leftarrow \mathsf{E.Enc}\big(\mathsf{pk}_\mathsf{E}, (b, \mathbf{y}, \mathbf{r}); r\big) \\ & \|\mathbf{y}\| \leq \beta \end{array} \right\}. \tag{6}$$

If the encryption scheme E is a lattice-based scheme, then one can use the general-purpose zero-knowledge proof system from [9], which also proposes an efficient zk-friendly hash function.

3.3 Proving Signature Rejection

We define the relation $R^{\perp}$ modeling that a vector $\boldsymbol{\sigma}$ did not pass the rejection sampling (Figure 2) for center $\boldsymbol{\sigma}_0$ and randomness u as follow:

$$R^{\perp} = \left\{ \begin{array}{l|l} \mathsf{st} = (c, \boldsymbol{\sigma}_0, u), & c = \mathsf{E.Enc}\big(\mathsf{pk}_{\mathsf{E}}, (b, \mathbf{y}, \mathbf{r}); r\big) \\ w = (r, b, \mathbf{y}, \mathbf{r}) & \|\mathbf{y}\| \leq \beta \\ & \boldsymbol{\sigma} = \mathbf{y} + (-1)^b \boldsymbol{\sigma}_0 \\ & \mathsf{Reject}(\boldsymbol{\sigma}, \boldsymbol{\sigma}_0, s; u) = \perp \end{array} \right\}, \tag{7}$$

where u is the randomness used in the rejection sampling.

We first describe how the randomness u in Reject is derived. Let F_0 be a hash function modeled as a random oracle. We set the randomness u to be $F_0(\mathsf{nc}, \mathsf{nc_S})$ the hash of a nonce nc, committed by the user before receiving the signer's nonce $\mathsf{nc_S}$, and $\mathsf{nc_S}$ (such that neither the user or signer can predict its value). If F_0 is modeled as a random oracle, then its output u is uniformly distributed and we can apply the rejection Lemma 5 to obtain the desired distribution for our blind signature. The user's nonce nc is added to preserve blindness, by making sure that the signer can only derive u once the user communicated its value $\mathsf{nc} \in \{0,1\}^{2\lambda}$. This ensures that the signer cannot compute u when the rejection sampling is accepted, and $\boldsymbol{\sigma}$ is revealed. Otherwise, an adversary could check for which centers $\boldsymbol{\sigma}_0$ a given $\boldsymbol{\sigma}$ is accepted or not. Note that the nonce nc must be committed to before receiving $\mathsf{nc_S}$, otherwise, a malicious user could always repeat rounds by testing several nonces, until finding one that makes the rejection sampling fail.

We now detail how to prove the last statement

$$\mathsf{Reject}(\boldsymbol{\sigma}, \boldsymbol{\sigma}_0, s; u) = \perp.$$

One can check that this comes down to the relation

$$|\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| > s^2 \cdot \operatorname{arccosh}\left(\frac{\exp\left(\|\boldsymbol{\sigma}_0\|^2/(2s^2)\right)}{uM} \right) =: \alpha.$$

The right hand side of the equation can be entirely computed by the signer, as it only depends on u and $\boldsymbol{\sigma}_0$. To prove that $|\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \alpha > 0$, since we also have the upper bound by Cauchy-Schwarz

$$|\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| = \left|\langle \mathbf{y} + (-1)^b \boldsymbol{\sigma}_0, \boldsymbol{\sigma}_0 \rangle\right| \leq (\beta + \beta_0)\beta_0,$$

it suffices to prove that $|\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \tilde{\alpha} \in \mathbb{N} \cup \{0\}$ for $\tilde{\alpha} = \lfloor \alpha \rfloor + 1$ is an integer bounded by $\tilde{\beta} = (\beta + \beta_0)\beta_0 - \lfloor \alpha \rfloor$. Thus, we can rewrite $R^\perp$ as follow

$$R^\perp = \left\{ \begin{array}{l} \mathsf{st} = (c, \boldsymbol{\sigma}_0, u, \tilde{\alpha}, \tilde{\beta}), \\ w = (r, b, \mathbf{y}, \mathbf{r}) \end{array} \middle| \begin{array}{l} c = \mathsf{E.Enc}\big(\mathsf{pk_E}, (b, \mathbf{y}, \mathbf{r}); r\big) \\ \|\mathbf{y}\| \leq \beta \\ \boldsymbol{\sigma} = \mathbf{y} + (-1)^b \boldsymbol{\sigma}_0 \\ |\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \tilde{\alpha} \geq 0 \\ |\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \tilde{\alpha} \leq \tilde{\beta} \end{array} \right\}. \tag{8}$$

The two last statements can be shown by proving that the binary decomposition of $x = |\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \tilde{\alpha}$ can be written in $\ell = \log(\tilde{\beta})$ bits. Let $\mathbf{G}$ be the gadget matrix $\mathbf{G} = (1, 2, \ldots, 2^{\ell-1})$ mapping a binary vector to its integer representation. Then, this requires to additionally commit to the binary vector $\mathbf{x} = (\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1})$, and to prove that

(1) $\mathbf{G}\mathbf{x} = |\langle \boldsymbol{\sigma}, \boldsymbol{\sigma}_0 \rangle| - \tilde{\alpha} \mod q$,
(2) $\langle \mathbf{x}, \mathbf{x} - \mathbf{1} \rangle = 0 \mod q$ (to prove that $\mathbf{x}$ is binary),
(3) $(\mathbf{x}, \boldsymbol{\sigma})$ is shorter than some bound, to ensure that the two equations above hold over the integers.

It results that this can be reduced to scalar product relations and a norm check, which fits relations handled by the proof system [9]. This method can be found in more details in [36][6]. In their case, they want to prove exact norm bounds of the form $\|\mathbf{A}\mathbf{s} - \mathbf{v}\| \leq \beta$, by proving that the value

$$\beta^2 - \langle \mathbf{A}\mathbf{s} - \mathbf{v}, \mathbf{A}\mathbf{s} - \mathbf{v} \rangle \geq 0$$

can be written as a positive integer of $2 \log \beta$ bits.

3.4 Resulting Proof System

We now define $\Pi = \{\mathsf{Prove}, \mathsf{Verif}\}$ the non-interactive ZKP system required by our blind signature protocol. In the first round, Π only proves the honest computation of a request while in the following rounds, it also proves that the previous value of $\boldsymbol{\sigma}$ was rejected. Since both relations are included in the same proof, the added overhead between the first and the following rounds can be mitigated (*e.g.,* as noted in [9], the proof size remains almost constant across use cases and input size in their benchmarks).

We define its algorithms:

- $\Pi.\mathsf{Prove}(m, r, r_m, b, \mathbf{y}, \mathbf{r}, \mathsf{prev_U})$: prove the relation R^{rq} (9) for the witnesses $(m, r, r_m, b, \mathbf{y}, \mathbf{r})$. If $\mathsf{prev_U} \neq \emptyset$ *i.e.,* if we are not in the first round, also prove the relation $R^\perp$ (9) with witnesses $\mathsf{prev_U} = (r_{\mathsf{prev}}, b_{\mathsf{prev}}, \mathbf{y}_{\mathsf{prev}}, \mathbf{r}_{\mathsf{prev}})$.
- $\Pi.\mathsf{Verif}(\mathsf{vk}, c_m, c, h, \mathsf{pnc}, \mathsf{prev_S}, \pi)$: verify if π is a valid proof for the relation R^{rq} for the state $\mathsf{st} = (\mathsf{vk}, c_m, c, h)$. If $\mathsf{prev_S} \neq \emptyset$ *i.e.,* if we are not in the first round, also verify $R^\perp$ for the state $(c_{\mathsf{prev}}, \boldsymbol{\sigma}_{0,\mathsf{prev}}, u)$ where $\mathsf{prev_S}$ is parsed as $(c_{\mathsf{prev}}, \boldsymbol{\sigma}_{0,\mathsf{prev}}, \mathsf{pnc_S})$ and $u = F_0(\mathsf{pnc}, \mathsf{pnc_S})$.

[6] See in Section 5.2, p.42, paragraph "General strategy".

4 Security Analysis

We show that our construction satisfies correctness, blindness and one-more unforgeability in the concurrent setting. We also prove that we can upper bound the average number of rounds even for malicious users.

Proof strategy. Since we work in the random oracle model, as discussed in [7], our proof of valid request requires proving the oracle-dependent relations

$$h_m = H(m, \rho),\ \rho = G(b, \mathbf{y}, \mathbf{r}),$$

which are not an NP relation in the random oracle model, as there is no circuit to evaluate. To address this issue, we first prove the one-more unforgeability of an unblinded version of our scheme $\mathsf{BS}^{\mathsf{UNF}}$ in the random oracle model, that does not require any ZKP. We assume that once the random oracles are instantiated with concrete hash functions, $\mathsf{BS}^{\mathsf{UNF}}$ remains unforgeable. Then, in the standard model, we prove that the unforgeability of our blind signature protocol BS can be reduced to the unforgeability of $\mathsf{BS}^{\mathsf{UNF}}$. Secondly, we prove blindness in the same fashion, by proving the blindness of another intermediate protocol $\mathsf{BS}^{\mathsf{BLIND}}$ in the random oracle model, that is not unforgeable and does not contain any ZKP. We assume that once the random oracles are instantiated with concrete hash functions, $\mathsf{BS}^{\mathsf{BLIND}}$ remains blind, and we prove that the blindness of our blind signature protocol BS can be reduced to this assumption. Thus, we have to make the two following assumptions:

- When the random oracles are instantiated with concrete hash functions, $\mathsf{BS}^{\mathsf{UNF}}$ remains one-more unforgeable.
- When the random oracles are instantiated with concrete hash functions, $\mathsf{BS}^{\mathsf{BLIND}}$ remains blind.

These are implied by standard assumptions when the hash functions are modeled as random oracles, which is strong evidence that these assumptions still hold whenever H and G are instantiated by secure concrete hash functions.

4.1 Correctness

Let δ_n be the discrete gaussian tail-cut parameter as defined in Lemma 4.

Theorem 1. *Let $\beta = \delta_n s\sqrt{2n}$ and $\beta_0 = \delta_n s_0\sqrt{2n}$, where $s = \gamma\beta_0$. Our blind-signature scheme from Figure 4 is correct, and terminates on average in M iterations for $M = \exp(1/(2\gamma^2))$.*

Proof. The proof is given in Appendix C.

4.2 Round Complexity for Malicious Users

Theorem 2. *Let* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{BS.KeyGen}(\lambda)$. *Let* F *be a collision resistant hash function, and* F_0 *be a hash function modeled as a random oracle. Assume that the proof system* Π *is sound. For any adversary* $\mathcal{A}$, $\mathsf{BS.Sign}\langle\mathcal{A}(\mathsf{vk}), \mathsf{Signer}(\mathsf{vk}, \mathsf{sk})\rangle$ *will terminate on average in less than* $\tilde{M}$ *rounds, where*

$$\tilde{M} = \exp\left(\frac{1+\delta_n^2}{2\delta_n^2\gamma^2} + \frac{\sqrt{2\pi}}{\gamma}\right).$$

Proof. The proof is given in Appendix D.

4.3 One-more Unforgeability

We start by proving the one-more unforgeability of an intermediate protocol $\mathsf{BS}^{\mathsf{UNF}}$ (Figures 6 and 5) in the random oracle model. This protocol does not contain any ZKP, as the user sends its message and randomness in plaintext.

Theorem 3. *We model* G *and* H *as random oracles. Let* $s_0 \geq \eta_\varepsilon(\mathbb{Z}^{2n})\|\tilde{\mathbf{T}}\|$ *such that* $s_0 > \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$, *and* $s > s_0$. *Then, the protocol* $\mathsf{BS}^{\mathsf{UNF}}$ *is one-more unforgeable under the* NTRU *and* MSIS *assumptions, and the pseudorandomness of the* PRF *family* $\mathcal{F}$. *More precisely, for* k *the number of concurrent signing queries and* n_q *the number of random oracle queries, we have*

$$\mathsf{Adv}_{\mathsf{UNF}}(\mathcal{A}, k) \leq \mathsf{Adv}_{\mathsf{MSIS}_B}(\mathcal{A}) + \mathsf{Adv}_{\mathsf{NTRU}}(\mathcal{A}) + kM\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{A}) + \frac{n_q^2}{2^\lambda} + n_q\frac{\varepsilon}{2},$$

for $B = 2\beta + \beta_0$ *and* M *the maximum number of repetitions of the signing protocol in the experiment.*

Proof. We prove this theorem by a sequence of games.

$\underline{\mathsf{G}_0}$: This is the original one-more unforgeability game as in Figure 1.

$\underline{\mathsf{G}_1}$: The same as G_0, except that the challenger simulates the random oracle G by keeping track of the previous queries in a list $\mathcal{L}_G$. On input a tuple $(b, \mathbf{y}, \mathbf{r})$ already queried, the challenger answers consistently. Otherwise, it samples $\rho \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$, adds $(\rho, b, \mathbf{y}, \mathbf{r})$ to $\mathcal{L}_G$ and outputs ρ. This simulates perfectly the random oracle G, hence we have $\mathsf{Adv}_0(\mathcal{A}) = \mathsf{Adv}_1(\mathcal{A})$.

$\underline{\mathsf{G}_2}$: The same as G_1, except that the challenger simulates the random oracle H by keeping track of the previous queries with two lists, $\mathcal{L}_H$ and $\mathcal{L}_{\mathsf{resp}}$, in the following way. On input a tuple (m, ρ) already queried, the challenger answers consistently. Otherwise if $(\rho, \cdot, \cdot, \cdot) \notin \mathcal{L}_G$, sample $\boldsymbol{\sigma} \leftarrow\!\!\$\ D_s^2$. If there is a tuple $(\rho, b, \mathbf{y}, \mathbf{r}) \in \mathcal{L}_G$, sample $\boldsymbol{\sigma}_0 \leftarrow\!\!\$\ D_{s_0}^2$. While $\|\boldsymbol{\sigma}_0\| > \beta_0$, resample. Then, set $\boldsymbol{\sigma} = \mathbf{y} + (-1)^b\boldsymbol{\sigma}_0$. In both cases, return

$$h_m = \mathbf{A}\boldsymbol{\sigma} \mod q,$$

and add $(\boldsymbol{\sigma}, m, \rho)$ to $\mathcal{L}_H$. In the latter case, also save the preimage $(\boldsymbol{\sigma}_0, m, \rho)$ in $\mathcal{L}_{\mathsf{resp}}$. Since $s_0 \geq \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$, we know from Lemma 1 that the distribution

$\mathsf{BS}^{\mathsf{UNF}}.\mathsf{KeyGen}(\lambda)$:

$\kappa \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$(\mathbf{T}, \mathbf{A}) \leftarrow \mathsf{TrapGen}(q)$
$\mathsf{vk} = \mathbf{A}, \mathsf{sk} = (\mathbf{T}, \kappa)$
return $(\mathsf{vk}, \mathsf{sk})$

$\mathsf{BS}^{\mathsf{UNF}}.\mathsf{GenRq}(\mathsf{vk}, m, \mathsf{pnc})$:

$b \leftarrow\!\!\$\ \{0,1\}$
$\mathbf{y} \leftarrow\!\!\$\ D_s^2$
$\mathbf{r} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
$c_{\mathsf{nc}} \leftarrow F(\mathsf{nc})$
if first round:
  $\mathsf{rq} \leftarrow (m, b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}})$
else $\mathsf{rq} \leftarrow (b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}}, \mathsf{pnc})$
return $(\mathsf{rq}, b, \mathbf{y}, \mathbf{r}, \mathsf{nc})$

$\mathsf{BS}^{\mathsf{UNF}}.\mathsf{Verif}(\mathsf{vk}, m, (\boldsymbol{\sigma}, \rho))$:

if $\big(\mathbf{A}\boldsymbol{\sigma} = H(m, \rho) \mod q\big)$
  $\wedge\ (\|\boldsymbol{\sigma}\| \leq \beta)$:
  return $\top$
return $\bot$

$\mathsf{BS}^{\mathsf{UNF}}.\mathsf{GenPSig}(\mathsf{vk}, \mathsf{sk}, \mathsf{rq}, c_{\mathsf{pnc}}, \mathsf{prev}_{\mathsf{S}})$:

if first round:
  Parse rq as $(m, b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}})$
else :
  Parse rq as $(b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}}, \mathsf{pnc})$
  if $F(\mathsf{pnc}) \neq c_{\mathsf{pnc}}$: **return** $\bot$
  Parse $\mathsf{prev}_{\mathsf{S}}$ as $(\boldsymbol{\sigma}_{\mathsf{prev}}, \boldsymbol{\sigma}_{0,\mathsf{prev}}, \mathsf{pnc}_{\mathsf{S}})$
  $u \leftarrow F_0(\mathsf{pnc}, \mathsf{pnc}_{\mathsf{S}})$
  if $\mathsf{Reject}(\boldsymbol{\sigma}_{\mathsf{prev}}, \boldsymbol{\sigma}_{0,\mathsf{prev}}, s; u) \neq \bot$:
    return $\bot$
if $\|\mathbf{y}\| > \beta$: **return** $\bot$
$\rho \leftarrow G(b, \mathbf{y}, \mathbf{r})$
$h \leftarrow (-1)^b (H(m, \rho) - \mathbf{A}\mathbf{y})$
$\mathsf{count} \leftarrow 0$
$k \leftarrow F_\kappa(h || \mathsf{count})$
$\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
while $\|\boldsymbol{\sigma}_0\| > \beta_0$:
  $\mathsf{count} \leftarrow \mathsf{count} + 1$
  $k \leftarrow F_\kappa(h || \mathsf{count})$
  $\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
$\mathsf{nc}_{\mathsf{S}} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
return $\mathsf{psig} = (\boldsymbol{\sigma}_0, \mathsf{nc}_{\mathsf{S}})$

Fig. 5: The KeyGen and Verif algorithms for the intermediate protocol $\mathsf{BS}^{\mathsf{UNF}}$.

of $\mathbf{A}\boldsymbol{\sigma}_0 \mod q$, hence also the distribution of $h_m \in R_q$, is at distance $\varepsilon/2$ from the uniform distribution. The same argument is used, but with $s > s_0$, for $h = \mathbf{A}\boldsymbol{\sigma} \bmod q$ if $\boldsymbol{\sigma}$ is sampled directly from D_s^2 when ρ does not appear in $\mathcal{L}_G$. It results that we have at most

$$|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq n_q \frac{\varepsilon}{2}.$$

Reduction to MSIS. We now build the following reduction $\mathcal{R}$ to the MSIS assumption. Let $\mathcal{A}$ be an adversary against the one-more unforgeability in G_2 with advantage $\varepsilon_{\mathcal{A}}$. The reduction takes as input an MSIS instance $\mathbf{A}$.

The reduction sends the public key $\mathsf{vk} = \mathbf{A}$ to $\mathcal{A}$. The adversary has a negligible advantage to distinguish this from an honest setup under the NTRU assumption. The reduction simulates the Signer without any trapdoor for k concurrent executions in the following way:

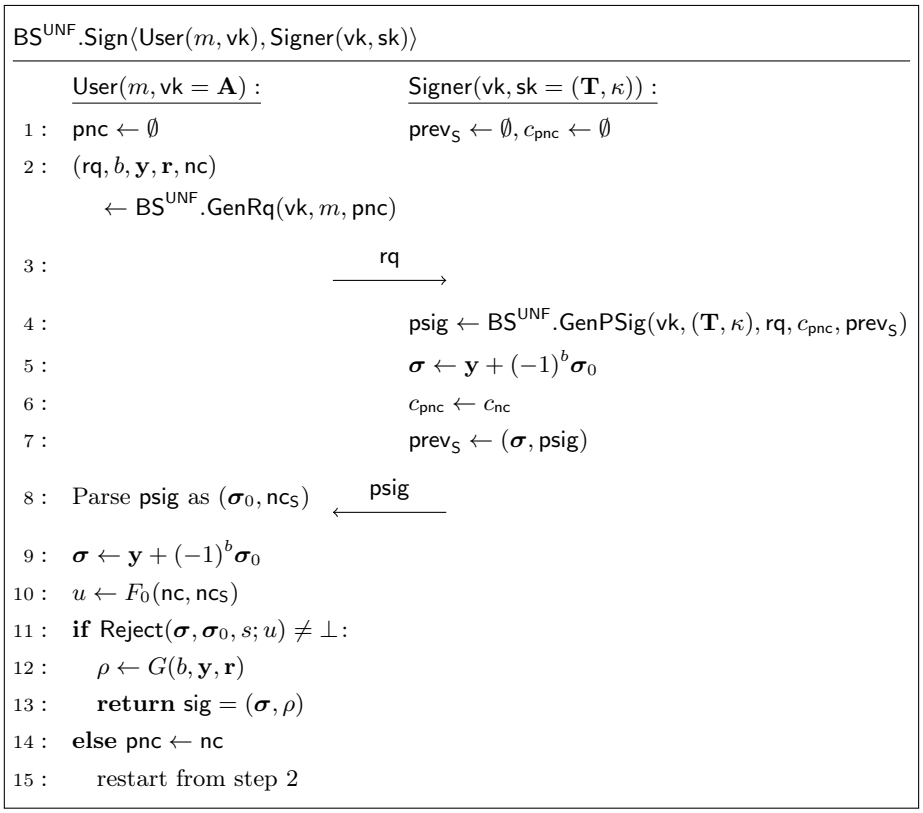

Fig. 6: Signing protocol for our intermediate scheme $\mathsf{BS}^{\mathsf{UNF}}$.

– Signer: for the i-th execution, parse an initial request rq in the first round as $(m,b,\mathbf{y},\mathbf{r},c_{\mathsf{nc}})$. The reduction checks that $\|\mathbf{y}\|\leq\beta$, and computes h from $\rho=G(b,\mathbf{y},\mathbf{r})$ as

$$h=(-1)^b(H(m,\rho)-\mathbf{A}\mathbf{y})\mod q.$$

If h was already queried in another parallel execution $j\in[k]$, return the corresponding $\boldsymbol{\sigma}_0$ consistently (as we required $\mathsf{SamplePre}$ to be deterministic). A tuple $(\boldsymbol{\sigma}_0,m,\rho)$ must appear in $\mathcal{L}_{\mathsf{resp}}$ since both G and H have been called, otherwise return $\perp$. Sample $\mathsf{nc}_\mathsf{S}\leftarrow\!\$\ \{0,1\}^{2\lambda}$ and output the pre-signature $\mathsf{psig}=(\boldsymbol{\sigma}_0,\mathsf{nc}_\mathsf{S})$.
If the i-th execution needs more rounds, parse the following requests rq as $(b,\mathbf{y},\mathbf{r},c_{\mathsf{nc}},\mathsf{pnc})$. First, we check that $c_{\mathsf{pnc}}=F(\mathsf{pnc})$, and that the previous blind signature was rejected. From $\mathsf{prev}_\mathsf{S}=(\boldsymbol{\sigma}_{\mathsf{prev}},\boldsymbol{\sigma}_{0,\mathsf{prev}},\mathsf{pnc}_\mathsf{S})$ and pnc, the reduction derives the randomness $u\leftarrow F_0(\mathsf{pnc},\mathsf{pnc}_\mathsf{S})$, and checks that

$$\mathsf{Reject}(\boldsymbol{\sigma}_{\mathsf{prev}},\boldsymbol{\sigma}_{0,\mathsf{prev}},s;u)=\perp.$$

If this is not verified, return $\perp$. Otherwise, proceed as in the first round with the new randomness and same value of m.

We show that the adversary cannot distinguish between our simulation of $\boldsymbol{\sigma}_0$ and a real signer's answer. The adversary's inputs are completely independent of the PRF key κ, thus under the pseudorandomness of the PRF family $\mathcal{F}$, the adversary cannot distinguish between $F_\kappa(h)$ and $f(h)$, for f a random function. Since we have $s_0 \geq \eta_\varepsilon(\mathbb{Z}^{2n})\|\tilde{\mathbf{T}}\|$ as in Lemma 2, SamplePre produces outputs with a distribution indistinguishable from $D_{\Lambda_q^h(\mathbf{A}),s_0}$ on input a random seed. Then, the adversary's advantage to distinguish our simulation of $\boldsymbol{\sigma}_0$ and a real one is at most $kM\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{A})$.

Otherwise, the reduction aborting a signing query that should have been valid can result from three events:

(1) $\mathcal{A}$ found a collision in H,
(2) $\mathcal{A}$ found a collision in G
(3) $\mathcal{A}$ asked for $H(m,\rho)$ before ρ being output by G.

Since $|R_q| = q^n \gg 2^\lambda$, the probability of these events is bounded by

$$\frac{n_q^2}{2|R_q|} + \frac{n_q^2}{2^{\lambda+1}} + \frac{n_q}{2^\lambda} \leq \frac{n_q^2}{2^\lambda}.$$

Exploiting the forgery. Finally, when the adversary outputs a forgery of $k+1$ valid signatures

$$\{\big(m_i, (\boldsymbol{\sigma}_i, \rho_i)\big) \mid i \in [k+1]\}$$

for only k signing queries, the adversary looks for a message $m^\star$ that was not used in the signing queries. Indeed, since whenever the user fails to derive a blind signature, it has to continue the signing execution with the same message, the user could only make queries on at most k different messages. For the corresponding signature $(\boldsymbol{\sigma}^\star, \rho^\star)$, the adversary must have queried the random oracle H to get $h_{m^\star} = H(m^\star, \rho^\star)$. Otherwise, the probability to guess it randomly is $1/|R_q| < 2^{-\lambda}$. Hence, with overwhelming probability, there exists an entry $(\boldsymbol{\sigma}, m^\star, \rho^\star) \in \mathcal{L}_H$ such that we have $\mathbf{A}(\boldsymbol{\sigma}^\star - \boldsymbol{\sigma}) = 0 \mod q$. The probability that this is not a valid MSIS solution because $\boldsymbol{\sigma}^\star - \boldsymbol{\sigma} = 0$ is negligible, since $\boldsymbol{\sigma}$ was chosen by the reduction following a discrete gaussian distribution independently of the adversary (as $m^\star$ was never in a signing query). Indeed, assume first that there exists some tuple $(\rho^\star, b, \mathbf{y}, \mathbf{r}) \in \mathcal{L}_G$ such that the reduction computed $\boldsymbol{\sigma}$ as $\boldsymbol{\sigma} = \mathbf{y} + (-1)^b\boldsymbol{\sigma}_0$ in the H oracle query. Given $h_{m^\star} \in R_q$, the conditioned distribution of $\boldsymbol{\sigma}_0$ is the discrete gaussian distribution with standard deviation s_0 over the coset

$$\Lambda_q^t(\mathbf{A}), \quad t = (-1)^b(h_{m^\star} - \mathbf{A}\mathbf{y}).$$

From Corollary 1, we know that $D_{\Lambda_q^t(\mathbf{A}),s_0}$ has around $n\log(s_0^2/q)$ bits of entropy, thus the probability that the reduction sampled $\boldsymbol{\sigma}_0$ such that $\boldsymbol{\sigma}^\star = \mathbf{y} + (-1)^b\boldsymbol{\sigma}_0$

is negligible[7]. Otherwise, if $\rho^\star$ is not an output of the random oracle G, the same argument applies this time with $\boldsymbol{\sigma}$ directly sampled by the reduction with standard deviation $s > s_0$ over the lattice coset $\Lambda_q^{h_{m^\star}}(\mathbf{A})$. Since $\|\boldsymbol{\sigma}^\star - \boldsymbol{\sigma}\|$ is at most $\beta + (\beta + \beta_0)$, the reduction solves the MSIS instance $\mathbf{A}$ for a bound $B = 2\beta + \beta_0$ with probability at least

$$p \geq \varepsilon_{\mathcal{A}} - \mathsf{Adv}_{\mathsf{NTRU}}(\mathcal{A}) - \frac{n_q^2}{2^\lambda} - kM\mathsf{Adv}_{\mathsf{PRF}}(\mathcal{A}).$$

The theorem follows by chaining the inequalities. □

Now we instantiate the random oracles in $\mathsf{BS}^{\mathsf{UNF}}$ with concrete hash functions, and assume it still achieves one-more unforgeability.

Theorem 4. *We instantiate both G and H with concrete hash functions. Our blind-signature scheme BS is one-more unforgeable under the assumptions that $\mathsf{BS}^{\mathsf{UNF}}$ instantiated with G and H is one-more unforgeable, and the soundness of Π. More precisely, for k the number of concurrent signing queries, we have*

$$\mathsf{Adv}^{\mathsf{BS}}_{\mathsf{UNF}}(\mathcal{A}, k) \leq \mathsf{Adv}^{\mathsf{BS}^{\mathsf{UNF}}}_{\mathsf{UNF}}(\mathcal{A}, k) + kM \cdot \mathsf{Adv}^{\Pi}_{\mathsf{SND}}(\mathcal{A})$$

for M the maximum number of repetitions of the signing protocol in the experiment.

Proof. We build the following reduction $\mathcal{R}$ to the one-more unforgeability of $\mathsf{BS}^{\mathsf{UNF}}$. Let $\mathcal{A}$ be an adversary against the one-more unforgeability of our blind signature scheme BS with advantage ε. The reduction takes as input the public key $\mathbf{A}$. It generates an encryption key $\mathsf{pk_E}$ but does not delete the associated secret key $\mathsf{sk_E}$, and sends $\mathsf{vk} = (\mathbf{A}, \mathsf{pk_E})$ to $\mathcal{A}$. This is identical to an honest setup. The reduction simulates the Signer for k concurrent executions by using the signing oracle for $\mathsf{BS}^{\mathsf{UNF}}$ in the following way:

– Signer: for the i-th execution, parse an initial request (first round of an execution) rq as $(h, c, \pi, c_{\mathsf{nc}}, c_m)$. If π is not a valid proof, return $\perp$. If h was already queried in another parallel execution $j \in [k]$, return the corresponding $\boldsymbol{\sigma}_0$ consistently. Otherwise, decrypt the ciphertexts as

$$m \leftarrow \mathsf{Dec}(c_m, \mathsf{sk_E}) \text{ and } (b, \mathbf{y}, \mathbf{r}) \leftarrow \mathsf{Dec}(c, \mathsf{sk_E})$$

and start a new signing protocol with $\mathsf{rq} = (m, b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}})$ to the signing oracle of $\mathsf{BS}^{\mathsf{UNF}}$. Forward the oracle answer psig to $\mathcal{A}$.
If the i-th execution needs more rounds, set $c_{\mathsf{pnc}} \leftarrow c_{\mathsf{nc}}$ and parse the following requests rq as $(h, c, \pi, c_{\mathsf{nc}}, \mathsf{pnc})$. Check if $c_{\mathsf{pnc}} = F(\mathsf{pnc})$, otherwise return $\perp$. Decrypt $(b, \mathbf{y}, \mathbf{r}) \leftarrow \mathsf{Dec}(c, \mathsf{sk_E})$ as before, and continue the same signing execution with the signing oracle of $\mathsf{BS}^{\mathsf{UNF}}$, by sending the new request $\mathsf{rq} = (b, \mathbf{y}, \mathbf{r}, c_{\mathsf{nc}}, \mathsf{pnc})$.

[7] For an NTRU lattice we have $\det(\Lambda_q^\perp(\mathbf{A})) = q^n$, and in instantiation, the requirement $s_0 \geq \eta_\varepsilon(\mathbb{Z}^{2n})\|\tilde{\mathbf{T}}\|$ yields $s_0 \approx 1.5\sqrt{q}$.

This simulates perfectly an honest signer, except if the adversary proved a false statement and $(m, b, \mathbf{y}, \mathbf{r})$ were not the real values used to compute h. The probability of this event is bounded by $kM \cdot \mathsf{Adv}^{\Pi}_{\mathsf{SND}}(\mathcal{A})$, since each signing execution takes at most M rounds.

Main reduction. Finally, when the adversary outputs a forgery of $k+1$ valid signatures

$$\{\big(m_i, (\boldsymbol{\sigma}_i, \rho_i)\big) \mid i \in [k+1]\}$$

for only k signing queries, the adversary looks for a message $m^\star$ that was not used in the signing queries. Since whenever the user fails to derive a signature, it has to prove that its new signing request still contains the same message, the user could only make signing queries on at most k different messages (otherwise, the protocol would have aborted as the reduction could not have simulated the signer correctly). Thus under the soundness of Π, there must be at least one $m^\star$ that was not in a signing query. Then, with overwhelming probability, the reduction obtains a forged signature $(m^\star, (\boldsymbol{\sigma}^\star, \rho^\star))$ that was not used in the signing queries, breaking the one-more unforgeability of $\mathsf{BS}^{\mathsf{UNF}}$, with probability at least $p \geq \varepsilon - kM \cdot \mathsf{Adv}^{\Pi}_{\mathsf{SND}}(\mathcal{A})$, and the theorem follows. □

4.4 Blindness

Next, we prove the blindness of an intermediate protocol $\mathsf{BS}^{\mathsf{BLIND}}$ (Figures 7 and 8) in the random oracle model. It is almost identical to BS, except that the user does not prove that its request is well-formed (thus, this scheme is not unforgeable), and does not rely on any ZKP. The key generation and verification algorithms are exactly the same as for BS (Figure 3).

Theorem 5. *We model F_0, F, G and H as random oracles. Let $\gamma > 1$, $\beta_0 = \delta_n s_0 \sqrt{2n}$ and $s = \gamma\beta_0 \geq \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$. The scheme $\mathsf{BS}^{\mathsf{BLIND}}$ is blind under the $\mathsf{IND\text{-}CPA}$ property of the encryption E. More precisely, we have*

$$\mathsf{Adv}_{\mathsf{BLIND}}(\mathcal{A}) \leq 2M \cdot \mathsf{Adv}_{\mathsf{IND\text{-}CPA}}(\mathcal{A}) + \frac{3M}{2^\lambda} + \varepsilon,$$

for M the maximum number of rounds in the two signing executions.

Proof. We prove this theorem by a sequence of games. We assume that both signatures do not abort. Each of the modifications that follow are performed twice in the same way in the signing protocols of both m_i, for each $i \in \{0, 1\}$.

$\underline{\mathsf{G}_0}$: This is the original blindness game as in Figure 1.

$\underline{\mathsf{G}_1}$: The same as G_0, except that the challenger replaces the ciphertexts with random values. Thus, we have

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq 2M \cdot \mathsf{Adv}_{\mathsf{IND\text{-}CPA}}(\mathcal{A}).$$

$\underline{\mathsf{G}_2}$: The same as above, except that the challenger chooses $\rho_i \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$ for $i \in \{0, 1\}$ at random, independently of the inputs $b_i, \mathbf{y}_i$ and $\mathbf{r}_i$. The adversary

$\mathsf{BS}^{\mathsf{BLIND}}.\mathsf{GenRq}(\mathsf{vk}, m, \mathsf{pnc})$:

1 : $\mathbf{y} \leftarrow\!\!\$\ D_s^2$
2 : $b \leftarrow\!\!\$\ \{0,1\}$
3 : $\mathbf{r} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
4 : $\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
5 : $c_{\mathsf{nc}} \leftarrow F(\mathsf{nc})$
6 : $h = (-1)^b(H(m, G(b, \mathbf{y}, \mathbf{r})) - \mathbf{A}\mathbf{y})$
7 : $c \leftarrow \mathsf{E.Enc}\big(\mathsf{pk}_{\mathsf{E}}, (b, \mathbf{y}, \mathbf{r}); r\big)$
8 : **if** first round: $\mathsf{rq} \leftarrow (h, c, c_{\mathsf{nc}}, c_m)$
9 : **else** $\mathsf{rq} \leftarrow (h, c, c_{\mathsf{nc}}, \mathsf{pnc})$
10 : **return** $(\mathsf{rq}, b, \mathbf{y}, \mathbf{r}, \mathsf{nc})$

$\mathsf{BS}^{\mathsf{BLIND}}.\mathsf{GenPSig}(\mathsf{vk}, \mathsf{sk} = (\mathbf{T}, \kappa), \mathsf{rq}, c_{\mathsf{pnc}})$:

1 : **if** first round:
2 : Parse rq as $(h, c, c_{\mathsf{nc}}, c_m)$
3 : **else** Parse rq as $(h, c, c_{\mathsf{nc}}, \mathsf{pnc})$
4 : **if** $F(\mathsf{pnc}) \neq c_{\mathsf{pnc}}$:
5 : **return** $\perp$
6 : $\mathsf{count} \leftarrow 0$
7 : $k \leftarrow F_\kappa(h||\mathsf{count})$
8 : $\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
9 : **while** $\|\boldsymbol{\sigma}_0\| > \beta_0$:
10 : $\mathsf{count} \leftarrow \mathsf{count} + 1$
11 : $k \leftarrow F_\kappa(h||\mathsf{count})$
12 : $\boldsymbol{\sigma}_0 \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, h, s_0; k)$
13 : $\mathsf{nc_S} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$
14 : **return** $\mathsf{psig} = (\boldsymbol{\sigma}_0, \mathsf{nc_S})$

Fig. 7: The GenRq and GenPSig subroutines for the protocol $\mathsf{BS}^{\mathsf{BLIND}}$.

cannot distinguish from the previous game unless it queries the random oracle G on $(b_i, \mathbf{y}_i, \mathbf{r}_i)$, which happens with probability $2^{-\lambda}$ since the adversary's inputs are completely independent of $\mathbf{r}_i \in \{0,1\}^{2\lambda}$. Thus, we have

$$|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq 2^{-\lambda}M.$$

$\underline{\mathsf{G}_3}$: The same as above, except that in each round, the challenger samples $c_{\mathsf{nc}} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$ at random and, after receiving the adversary's answer psig, samples $\mathsf{nc} \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$ and sets $F(\mathsf{nc}) = c_{\mathsf{nc}}$. This simulates F perfectly, except if the adversary already queried F on nc, which happens with probability at most $2^{-\lambda}$. Thus, we have

$$|\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A})| \leq 2^{-\lambda}M.$$

$\underline{\mathsf{G}_4}$: The same as above except that when computing a new request, the challenger samples $u_i \leftarrow\!\!\$\ R_q$ and programs the random oracle H such that $H(m_i, \rho_i) = u_i$ for all $i \in \{0,1\}$. Then, it computes

$$h_i = (-1)^{b_i}(u_i - \mathbf{A}\mathbf{y}_i) \mod q.$$

Since we have

$$\mathbf{A}\boldsymbol{\sigma}_i = \mathbf{A}\mathbf{y}_i + (-1)^{b_i}h_i = \mathbf{A}\mathbf{y}_i + (u_i - \mathbf{A}\mathbf{y}_i) = u_i \mod q,$$

this is consistent with the verification $H(m_i, \rho_i) = \mathbf{A}\boldsymbol{\sigma}_i \mod q$. Since this is only reordering the operations, this game is perfectly indistinguishable from G_2 and we have $\mathsf{Adv}_3(\mathcal{A}) - \mathsf{Adv}_4(\mathcal{A}) = 0$.

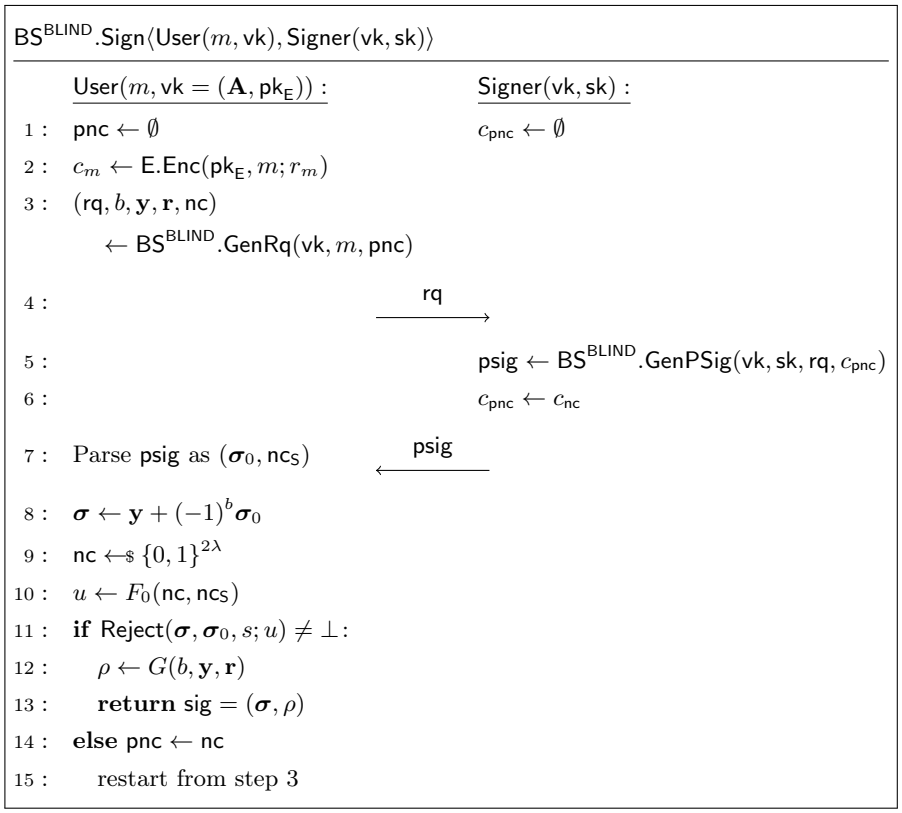

Fig. 8: The signing protocol for our intermediate scheme $\mathsf{BS}^{\mathsf{BLIND}}$.

$\underline{\mathsf{G}_5}$: The same as above, but the challenger samples $h_i \leftarrow\!\!\$\ R_q$ independently. Then, it programs the random oracle H such that $H(m_i, \rho_i) = u_i$, where $u_i := \mathbf{A}\mathbf{y}_i + (-1)^{b_i} h_i \mod q$ for $i \in \{0,1\}$. It results that

$$H(m_i, \rho_i) = \mathbf{A}\mathbf{y}_i + (-1)^{b_i} h_i = \mathbf{A}(\mathbf{y}_i + (-1)^{b_i}\boldsymbol{\sigma}_0) = \mathbf{A}\boldsymbol{\sigma}_i \mod q,$$

such that correctness is maintained. If the blind signature is accepted and the hash $H(m_i, \rho_i)$ is revealed, the indistinguishability between these games does not rely on one-time pad but rather on the fact that

$$h_i \longmapsto u_i := \mathbf{A}\mathbf{y}_i + (-1)^{b_i} h_i$$

is a bijection of $R_q \longrightarrow R_q$ of inverse

$$u_i \longmapsto h_i := (-1)^{b_i}(u_i - \mathbf{A}\mathbf{y}_i).$$

Thus, sampling uniformly $u_i =: H(m_i, \rho_i)$ first or h_i first yields the same exact joint distribution $(h_i, H(m_i, \rho_i))$.

Otherwise, if the blind signature is rejected, the adversary is not given the hash $H(m_i, \rho_i)$, but only the nonce to derive the randomness used in the rejection sampling (which can leak some information on the mask $\mathbf{y}_i$). Then, sampling $h_i \leftarrow\!\!\$\ R_q$ simulates perfectly

$$h_i = (-1)^{b_i}(H(m_i, \rho_i) - \mathbf{A}\mathbf{y}_i) \mod q$$

since the adversary does not know the random oracle value $H(m_i, \rho_i)$, except if the adversary queried the random oracle H on the exact input (m_i, ρ_i) in one of the failed rounds. This happens with probability at most $2^{-\lambda}$ in each round, since the adversary's inputs are completely independent of $\rho_i \in \{0,1\}^{2\lambda}$. Thus, the adversary's advantage to distinguish between these games is at most

$$|\mathsf{Adv}_4(\mathcal{A}) - \mathsf{Adv}_5(\mathcal{A})| \leq 2^{-\lambda}(M-1).$$

$\underline{\mathsf{G}_6}$: The same as above, but the challenger samples $b_i \leftarrow\!\!\$\ \{0,1\}$, $\mathbf{y}_i \leftarrow\!\!\$\ D_s^2$, and programs $H(m_i, \rho_i) = u_i = \mathbf{A}\mathbf{y}_i + (-1)^{b_i} h_i \mod q$ only *after* the adversary sends $\boldsymbol{\sigma}_{0,i}$. Indeed at this point, the requests h_i are sampled independently of the bits b_i, randomness $\mathbf{y}_i$, as well as $H(m_i, \rho_i)$. Therefore both the bit b_i and randomness $\mathbf{y}_i$ can be sampled after seeing the adversary's answers $\boldsymbol{\sigma}_{0,i}$, exactly as in classical bimodal rejection sampling. The only case where this is not possible is if the adversary queries the random oracle H on input (m_i, ρ_i) before sending $\boldsymbol{\sigma}_{0,i}$. Since the adversary's inputs are independent of ρ_i at this point, this can only happen if the adversary guesses $\rho_i \in \{0,1\}^{2\lambda}$. Thus, we have

$$|\mathsf{Adv}_5(\mathcal{A}) - \mathsf{Adv}_6(\mathcal{A})| \leq 2^{-\lambda}.$$

$\underline{\mathsf{G}_7}$: The same as above, but the challenger samples $\boldsymbol{\sigma}_i \leftarrow\!\!\$\ D_s^2$ for $i \in \{0,1\}$, and programs the random oracle H such that $H(m_i, \rho_i) = \mathbf{A}\boldsymbol{\sigma}_i \mod q$ for all $i \in \{0,1\}$. This is the only relation that needs to be maintained for $(\boldsymbol{\sigma}_i, \rho_i)$ to be a valid signature on m_i. The challenger still repeats rounds with probability $1/M$. From Lemma 5, we know that $\boldsymbol{\sigma}_i$ follows exactly the same distribution as an honest signature $\boldsymbol{\sigma}_i = \mathbf{y}_i + (-1)^{b_i}\boldsymbol{\sigma}_{0,i}$ due to the bimodal rejection sampling, now that we made sure that the request h_i is independent of $\mathbf{y}_i$ and b_i, and the bit b_i is only chosen after the adversary sends its response $\boldsymbol{\sigma}_{0,i}$. Note that since u is the output of a random oracle F_0 for fresh randomness nc, it is uniformly sampled, hence we can apply the rejection sampling lemma. Moreover, since $s \geq \eta_\varepsilon(\Lambda_q^\perp(\mathbf{A}))$, the distribution of $H(m_i, \rho_i)$ is at distance at most ε from the uniform distribution. Thus the adversary's advantage to distinguish between these games is at most

$$|\mathsf{Adv}_6(\mathcal{A}) - \mathsf{Adv}_7(\mathcal{A})| \leq \varepsilon.$$

Finally, we have $\mathsf{Adv}_7(\mathcal{A}) = 0$. In G_7, all the requests rq_i for $i \in \{0,1\}$ sent by the simulator in the signing process are completely independent of the message m_i that is being signed, and the output signature $(\boldsymbol{\sigma}_i, \rho_i)$ for $i \in \{0,1\}$ is computed independently from the signer's answer. Hence, the adversary cannot do better than a random guess. The theorem follows by chaining the inequalities. □

We can now prove the blindness of our blind signature scheme BS.

Theorem 6. *We instantiate the random oracles F_0, F, G and H with concrete hash functions. Our blind-signature scheme BS is blind under the assumptions that $\mathsf{BS}^{\mathsf{BLIND}}$ instantiated with G and H is blind, and the zero-knowledge property of Π. More precisely, we have*

$$\mathsf{Adv}^{\mathsf{BS}}_{\mathsf{BLIND}}(\mathcal{A}) \leq 2M \cdot \mathsf{Adv}^{\Pi}_{\mathsf{ZK}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{BS}^{\mathsf{BLIND}}}_{\mathsf{BLIND}}(\mathcal{A}),$$

for M the maximum number of rounds in the two signing executions.

Proof. We build a reduction $\mathcal{R}$ to the blindness of $\mathsf{BS}^{\mathsf{BLIND}}$. Let $\mathcal{C}$ be the challenger for the blindness game of $\mathsf{BS}^{\mathsf{BLIND}}$. Let $\mathcal{A}$ be an adversary against the blindness of BS with advantage ε.

Upon receiving two messages m_0, m_1 from $\mathcal{A}$, the reduction forwards them directly to $\mathcal{C}$. When $\mathcal{C}$ answers with a request $\mathsf{rq} = (h, c, c_{\mathsf{nc}}, c_m)$, the reduction uses the simulator of the ZKP to compute π a proof of validity for rq without the witnesses, and sends $\mathsf{rq}' = (h, c, c_{\mathsf{nc}}, \pi, c_m)$ to $\mathcal{A}$. Then, when the adversary answers with a pre-signature psig, the reduction forwards it to $\mathcal{C}$. If there are more rounds, the reduction keeps using the simulator of the ZKP to compute a valid request $\mathsf{rq}' \leftarrow (h, c, \pi, c_{\mathsf{nc}}, \mathsf{pnc})$ in BS from a request $\mathsf{rq} = (h, c, c_{\mathsf{nc}}, \mathsf{pnc})$ in $\mathsf{BS}^{\mathsf{BLIND}}$. The adversary's advantage to distinguish the reduction from an honest user is at most $2M \cdot \mathsf{Adv}^{\Pi}_{\mathsf{ZK}}(\mathcal{A})$, considering that each of the signing protocols for m_0 and m_1 takes at most M rounds.

Finally, when the challenger $\mathcal{C}$ outputs the blind signatures, the reduction forwards them to $\mathcal{A}$. When $\mathcal{A}$ outputs its guess b', the reduction answers the same to $\mathcal{C}$. Then, the reduction breaks the blindness of $\mathsf{BS}^{\mathsf{BLIND}}$ with advantage at least

$$\varepsilon - 2M \cdot \mathsf{Adv}^{\Pi}_{\mathsf{ZK}}(\mathcal{A}),$$

and the theorem follows. □

5 Parameter Selection

We propose some parameter selections for our scheme, for a core-SVP hardness respectively of between 116 bits and 148 bits[8]. Everything is summarized in Figure 2, while our exact parameter choices can be found in more details in Table 3. We provide a script to estimate the security of our selection at

`https://anonymous.4open.science/r/LatticeBlindSignature-DAA3/`

We consider the 3-smooth cyclotomic rings R_q of degree $n = 1296$ (resp. $n = 1536$), and prime modulus q. Since we use a bigger modulus than the usual Falcon moduli [22], we need to instantiate our NTRU trapdoor algorithms

[8] We note that in [7], the authors aim for a root Hermite factor $\delta < 1.0045$ which corresponds to a core-SVP hardness of 97 bits. In [1], the parameters are given for a core-SVP hardness of 109 bits.

with more general schemes such as MITAKA or ANTRAG [19,20]. They avoid floating point arithmetic in their sampler, which allows them to adapt better to bigger moduli, while having comparable results in terms of trapdoor quality. Moreover, the main advantage of those schemes compared to Falcon is that they are not limited to power-of-two cyclotomic rings, and work with the NTTRU rings [37,18] of the form

$$R = \mathbb{Z}[x]/(x^n - x^{n/2} + 1),$$

where n is a product of power-of-two and power-of-three (*i.e.,* with a 3-smooth conductor). We consider this type of rings as we could not reach a satisfying tradeoff with power-of-two dimensions: we reach only 90 bits of security for $n = 1024$ (which is no longer considered secure by the NIST), while we obtain a security level around 219 bits for $n = 2048$, which jumps directly to the NIST security level 4 (above 192 bits). To be able to rely on NTT (Number Theoretic Transform) for fast polynomial multiplication, we select primes such that $q \equiv 1 \mod m$, where m is such that $\phi(m) = n$. For these types of samplers, the standard deviation of the signer's pre-signature has to verify

$$s_0 \geq \eta_\varepsilon(\mathbb{Z}^{2n})\|\tilde{\mathbf{T}}\| \approx 1.5\sqrt{q},$$

to avoid leaking information on the trapdoor [20] (see Lemma 2). As for the bimodal rejection sampling (see Lemma 5), the standard deviation of the mask $\mathbf{y}$ has to be set at $s = \gamma\beta_0$. From Lemma 4, we can bound the norm of the resulting blind signature by $\beta = \delta_n s\sqrt{2n} = 2\delta_n^2\gamma s_0 n$.

MSIS *hardness.* We estimate the hardness of the MSIS instance for the bound $B = 2\beta + \beta_0$ in Appendix E. We obtain a security of respectively 116 bits and 148 bits, for our parameter selections of degree resp. $n = 1296$ and $n = 1536$.

Signature and transcript size. As in Falcon, we do not need to output the whole vector $\boldsymbol{\sigma} = (s_1, s_2) \in R_q^2$ but only $s_2 \in R_q$. Indeed s_1 can be recomputed as $s_1 = H(m) - s_2 h$, and one can check that the resulting (s_1, s_2) has small norm. Thus, the signature sig consists of $s_2 \in R_q$ and the hash $\rho \in \{0,1\}^{256}$, which results in a total of $n \cdot \log_2(q) + 256$ bits. As for the transcript, the size is greatly dominated by the size of the zero-knowledge proof in the request. One can use the succinct ZKP system of [9] that relies on both [8] and [36], which allows to prove knowledge of a cryptographic hash preimage in approximately 110 KB.

Efficiency. The recent work of [9] provided some benchmarks regarding the cost of the ZKP of the blind signature protocol of [7], where they found a prover time around 1.18s. Since our scheme requires proving almost the same statements, we can expect our prover time to be of the same order. Although the user-side cost is not satisfactory yet, we note that it is not too far from the ones from [30] for instance, that computes a signing request in around 0.5s (but yields 5 times bigger blind signatures). We believe that this can be a reasonable tradeoff in settings such as e-voting, where the primary objective is to minimize

computations on the signer side, and the blind signature sizes. Shifting most of the workload to the user, that executes the protocol only a few times, can then be acceptable to obtain shorter signatures.

Param.	**Instantiation**			**Description**
n	1296		1536	dimension of the ring R
λ	116	121	148	security parameter
M	≤ 1.1	≤ 1.1	≤ 1.1	$\mathbb{E}$(#rounds) for honest users
$\tilde{M}$	≤ 2.6	≤ 3.8	≤ 3.0	$\mathbb{E}$(#rounds) for malicious users
δ	< 1.0040	< 1.0039	< 1.0034	root Hermite factor
$\lvert\mathsf{sig}\rvert$	4.70 KB	4.54 KB	5.57 KB	blind signature size
$\lvert\mathsf{tr}\rvert$	110 KB			transcript size

Table 2: A parameter selection for our scheme. We propose two different sets of parameters, for ring dimensions $n = 1296 = 2^4 3^4$ and $n = 1536 = 2^9 3^1$. The prime modulus q is selected such that $q \equiv 1 \mod m$. We denote by $\mathbb{E}$(#rounds) the expected number of rounds. More details can be found in Table 3.

6 Conclusion

We propose a concurrently-secure solution based on the standard MSIS assumption, showing what can be gained when constructing a lattice-based blind signature scheme without relying, in the final blind signature, on general-purpose ZKP systems such as [36]. Our scheme is secure in the random oracle model. We describe our method following the structure of [7], which represents currently the most compact lattice-based signature scheme based on a standard assumption. By instantiating our protocol, we show that it yields significantly shorter blind signature sizes as small as 4.5 KB, which is more than four times smaller than the blind signatures of 22 KB obtained in [7]. Our signing protocol follows the steps of a classical Σ–protocol with bimodal rejection sampling [16]. As a tradeoff, our scheme is no longer round-optimal, with an average of 1.1 rounds for honest users. To preserve one-more unforgeability, a user must prove that it keeps the same message throughout the different rounds of a single execution. Users are also required to prove that they failed to derive a blind signature in the previous round, before starting a new round. We prove that this forces malicious users to terminate the protocol in at most 2.6 rounds on average (for 4.7 KB signatures, otherwise in 3.8 for 4.5 KB signatures). While we managed to remove the ZKP from the blind signature itself, there are still ZKPs appearing in the transcript, inside the signature request. As a result, our transcript size is still comparable to the one in [7], with a transcript of 120 KB on average. Exploring methods to reduce this component and remove the need for any ZKP in the protocol, would be an interesting direction for future work. Another interesting route would be to get rid of the random oracle model, and find a solution in the standard model while maintaining short blind signature sizes.

References

1. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Practical, round-optimal lattice-based blind signatures. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 39–53 (2022)
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Cryptology ePrint Archive, Paper 2015/046 (2015), `https://eprint.iacr.org/2015/046`
3. Alkeilani Alkadri, N., El Bansarkhani, R., Buchmann, J.: Blaze: practical lattice-based blind signatures for privacy-preserving applications. In: International Conference on Financial Cryptography and Data Security. pp. 484–502. Springer (2020)
4. Alkeilani Alkadri, N., El Bansarkhani, R., Buchmann, J.: On lattice-based interactive protocols: an approach with less or no aborts. In: Australasian Conference on Information Security and Privacy. pp. 41–61. Springer (2020)
5. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen **296**(1), 625–635 (1993)
6. Baum, C., Beckmann, M., Beullens, W., Mukherjee, S., Rechberger, C.: Concretely efficient blind signatures based on vole-in-the-head proofs and the mayo trapdoor. Cryptology ePrint Archive (2026)
7. Beullens, W., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Lattice-based blind signatures: Short, efficient, and round-optimal. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 16–29 (2023)
8. Beullens, W., Seiler, G.: Labrador: compact proofs for r1cs from module-sis. In: Annual International Cryptology Conference. pp. 518–548. Springer (2023)
9. Biasioli, B., Bolboceanu, M., Lyubashevsky, V., Merino-Gallardo, A., Osadnik, M., Seiler, G., Steuer, P.: A toolkit for succinct lattice-based zero knowledge proofs. Cryptology ePrint Archive (2026)
10. Bouaziz-Ermann, S., Canard, S., Eberhart, G., Kaim, G., Roux-Langlois, A., Traoré, J.: Lattice-based (partially) blind signature without restart. Cryptology ePrint Archive (2020)
11. Bouillaguet, C., Feneuil, T., Maire, J., Rivain, M., Sauvage, J., Vergnaud, D.: Blinding post-quantum hash-and-sign signatures. Cryptology ePrint Archive (2025)
12. Brands, S.: Untraceable off-line cash in wallet with observers. In: Annual international cryptology conference. pp. 302–318. Springer (1993)
13. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology: Proceedings of Crypto 82. pp. 199–203. Springer (1983)
14. Chen, Y.: Réduction de réseau et sécurité concrete du chiffrement completement homomorphe. Ph.D. thesis, Paris 7 (2013)
15. Davidson, A., Goldberg, I., Sullivan, N., Tankersley, G., Valsorda, F.: Privacy pass: Bypassing internet challenges anonymously. Proceedings on Privacy Enhancing Technologies (2018)
16. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Annual Cryptology Conference. pp. 40–56. Springer (2013)
17. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
18. Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient ntru instantiations. In: IACR International Conference on Public-Key Cryptography. pp. 65–94. Springer (2023)

19. Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., Yu, Y.: Mitaka: a simpler, parallelizable, maskable variant of falcon. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 222–253. Springer (2022)
20. Espitau, T., Nguyen, T.T.Q., Sun, C., Tibouchi, M., Wallet, A.: Antrag: Annular ntru trapdoor generation: Making mitaka as secure as falcon. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 3–36. Springer (2023)
21. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Annual International Cryptology Conference. pp. 60–77. Springer (2006)
22. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon, 2017. URL https://falcon-sign. info/. Accessed pp. 07–13 (2023)
23. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: International Workshop on the Theory and Application of Cryptographic Techniques. pp. 244–251. Springer (1992)
24. Genise, N., Micciancio, D., Peikert, C., Walter, M.: Improved discrete gaussian and subgaussian analysis for lattice cryptography. In: IACR International Conference on Public-Key Cryptography. pp. 623–651. Springer (2020)
25. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206 (2008)
26. Hanzlik, L., Lai, Y.F., Mula, M., Paracucchi, E., Slamanig, D., Tang, G.: Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum groups actions. Cryptology ePrint Archive (2025)
27. Hanzlik, L., Lai, Y.F., Paracucchi, E., Persichetti, E.: Wombat: Post-quantum blind signature from standard group action assumptions and more. Cryptology ePrint Archive (2026)
28. Hanzlik, L., Manulis, M., Mula, M., Pulval-Dady, A., Silde, T., Slamanig, D.: Braccoon: Concurrently secure blind lattice signatures from raccoon. Cryptology ePrint Archive (2026)
29. Hauck, E., Kiltz, E., Loss, J., Nguyen, N.K.: Lattice-based blind signatures, revisited. In: Annual International Cryptology Conference. pp. 500–529. Springer (2020)
30. Jeudy, C., Sanders, O.: Improved lattice blind signatures from recycled entropy. In: Annual International Cryptology Conference. pp. 477–513. Springer (2025)
31. Jeudy, C., Sanders, O.: Tag-friendly lattice sampler and applications. Cryptology ePrint Archive (2026)
32. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75**(3), 565–599 (2015)
33. Le, H.Q., Susilo, W., Khuc, T.X., Bui, M.K., Duong, D.H.: A blind signature from module lattices. In: 2019 IEEE Conference on Dependable and Secure Computing (DSC). pp. 1–8. IEEE (2019)
34. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
35. Lyubashevsky, V., Nguyen, N.K., Plancon, M.: Efficient lattice-based blind signatures via gaussian one-time signatures. In: IACR International Conference on Public-Key Cryptography. pp. 498–527. Springer (2022)

36. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general. In: Annual International Cryptology Conference. pp. 71–101. Springer (2022)
37. Lyubashevsky, V., Seiler, G.: Nttru: truly fast ntru using ntt. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 180–201 (2019)
38. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM journal on computing **37**(1), 267–302 (2007)
39. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Annual international cryptology conference. pp. 31–53. Springer (1992)
40. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Theory of Cryptography Conference. pp. 80–99. Springer (2006)
41. Papachristoudis, D., Hristu-Varsakelis, D., Baldimtsi, F., Stephanides, G.: Leakage-resilient lattice-based partially blind signatures. IET Information Security **13**(6), 670–684 (2019)
42. del Pino, R., Katsumata, S.: A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In: Annual International Cryptology Conference. pp. 306–336. Springer (2022)
43. del Pino, R., Katsumata, S., Prest, T., Rossi, M.: Raccoon: A masking-friendly signature proven in the probing model. In: Annual International Cryptology Conference. pp. 409–444. Springer (2024)
44. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of cryptology **13**(3), 361–396 (2000)
45. Rückert, M.: Lattice-based blind signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 413–430. Springer (2010)
46. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. cryptology eprint archive (2004)
47. Zhang, P., Jiang, H., Zheng, Z., Hu, P., Xu, Q.: A new post-quantum blind signature from lattice assumptions. IEEE Access **6**, 27251–27258 (2018)

A Useful Results

We need the following tail-bound for discrete gaussians from [5], which we use for $k = 2$ and $\delta = \delta_n$ such that

$$\delta^{2n} \exp\left(n(1-\delta^2)\right) \leq 2^{-\lambda}$$

in our protocol.

Lemma 4. *For any $\delta > 0$, we have*

$$\Pr[\|\mathbf{z}\| > \delta s\sqrt{kn} \mid \mathbf{z} \leftarrow\$ D_s^k] < \delta^{kn} \exp\left(\frac{kn}{2}(1-\delta^2)\right).$$

We use also this result, that we prove as a corollary of [24, Lemma 2.6].

Corollary 1. *Let Λ be a n-dimensional lattice, $0 \leq \varepsilon < 1$ and $\eta_\varepsilon(\Lambda) \leq s$. Then for any vector $\mathbf{c}$, we have*

$$H_\infty(D_{\Lambda+\mathbf{c},s}) \geq n \log s - \log \det(\Lambda) + \log(1-\varepsilon).$$

Proof. From [24, Lemma 2.6], we know that for any lattice Λ', $\mathbf{c} \in \mathsf{Span}(\Lambda')$, and $\varepsilon \geq 0$, where $\eta_\varepsilon(\Lambda') \leq 1$ we have

$$\rho(\Lambda' + \mathbf{c}) \in [1-\varepsilon, 1+\varepsilon] \cdot 1/\det(\Lambda').$$

We consider the scale $\Lambda' = (1/s)\Lambda$ of Λ, such that $\eta_\varepsilon(\Lambda') = \eta_\varepsilon((1/s)\Lambda) \leq 1$ by linearity. It yields that

$$\rho_s(\Lambda) = \rho(\Lambda') \in [1-\varepsilon, 1+\varepsilon] \cdot 1/\det(\Lambda') = [1-\varepsilon, 1+\varepsilon] \cdot s^n/\det(\Lambda),$$

where $\det(\Lambda') = (1/s^n)\det(\Lambda)$, since Λ has rank n. Let $\mathbf{c} = \mathbf{c}_\Lambda + \mathbf{c}_\perp$ where $\mathbf{c}_\perp \perp \mathsf{Span}(\Lambda)$ and $\mathbf{c}_\Lambda \in \mathsf{Span}(\Lambda)$. Then, we have

$$H_\infty(D_{\Lambda+\mathbf{c},s}) = -\log \max_{\mathbf{v}\in\Lambda+\mathbf{c}} \frac{\rho_s(\mathbf{v})}{\rho_s(\Lambda+\mathbf{c})} \geq -\log \frac{\rho_s(\mathbf{c}^\perp)}{\rho_s(\Lambda+\mathbf{c})}.$$

Since $\rho(\Lambda+\mathbf{c}) = \rho(\mathbf{c}_\perp)\rho(\Lambda+\mathbf{c}_\Lambda)$, we have

$$H_\infty(D_{\Lambda+\mathbf{c},s}) \geq \log \rho_s(\Lambda+\mathbf{c}_\Lambda).$$

It results that

$$H_\infty(D_{\Lambda+\mathbf{c},s}) \geq n \log s - \log \det(\Lambda) + \log(1-\varepsilon).$$

□

B Bounded Smoothing Parameter Proof

We detail here how to prove that a matrix $\mathbf{A}$ defines a lattice $\Lambda_q^\perp(\mathbf{A})$ with a sufficiently small smoothing parameter.

Let $\mathbf{A}$ be a matrix defining a n-dimensional lattice $\Lambda_q^\perp(\mathbf{A})$, and $\mathbf{T}$ be one of its bases. We know that, using the following bound from [25],

$$\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leq \|\tilde{\mathbf{T}}\| \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}},$$

bounding the smoothing parameter of $\Lambda_q^\perp(\mathbf{A})$ can be reduced to bounding the Gram-Schmidt norm $\|\tilde{\mathbf{T}}\|$ of $\mathbf{T}$. Let $p \nmid q$ be a public prime, $\mathbf{U}$ be the modular inverse of $\mathbf{T}$ modulo p, and let $\{\mathbf{t}_i\}_{i\in[n]}$ be the columns of $\mathbf{T}$. Then, one can prove the following bound on the smoothing parameter of $\Lambda_q^\perp(\mathbf{A})$

$$\eta_\varepsilon(\Lambda_q^\perp(\mathbf{A})) \leq \beta \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}$$

by proving the relation

$$R^{\eta} = \left\{ \begin{array}{l} \mathsf{st} = \mathbf{A}, \\ w = (\mathbf{T}, \mathbf{U}) \end{array} \middle| \begin{array}{l} \mathbf{AT} = 0 \mod q \\ \mathbf{TU} = \mathbf{I}_n \mod p \\ \forall i \in [n], \|\mathbf{t}_i\| \leq \beta \end{array} \right\}. \tag{9}$$

The inverse relation $\mathbf{TU} = \mathbf{I}_n \mod p$ proves that $\mathbf{T}$ has full rank. Let $\Lambda' \subseteq \Lambda_q^{\perp}(\mathbf{A})$ be the sublattice generated by $\mathbf{T}$, we then obtain that

$$\eta_{\varepsilon}(\Lambda_q^{\perp}(\mathbf{A})) \leq \eta_{\varepsilon}(\Lambda') \leq \beta \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}},$$

which is sufficient for our application. Then, proving this relation in zero-knowledge can be handled with the LNP proof system [36] or its more recent compressed version [9].

C Correctness

We provide here the full proof of Theorem 1.

Proof. The signer samples $\boldsymbol{\sigma}_0$ such that $\mathbf{A}\boldsymbol{\sigma}_0 = h = (-1)^b(H(m,\rho) - \mathbf{Ay})$. Hence, the final signature verifies

$$\mathbf{A}\boldsymbol{\sigma} = \mathbf{A}\big(\mathbf{y} + (-1)^b\boldsymbol{\sigma}_0\big) = \mathbf{Ay} + (-1)^b h = H(m,\rho) \mod q.$$

From the gaussian tail-bound in Lemma 4, the gaussian vectors $\mathbf{y}$ and $\boldsymbol{\sigma}_0$ sampled respectively with standard deviation s and s_0 verify the bounds

$$\|\mathbf{y}\| \leq \delta_n s\sqrt{2n} = \beta, \quad \|\boldsymbol{\sigma}_0\| \leq \delta_n s_0\sqrt{2n} = \beta_0,$$

with overwhelming probability. From Lemma 5, we know that the rejection sampling accepts $\boldsymbol{\sigma}$ with probability $1/M$ for $M = \exp(1/(2\gamma^2))$, and that the resulting distribution of $\boldsymbol{\sigma}$ is identical to D_s^2. Then, the gaussian tail bound also ensures that $\|\boldsymbol{\sigma}\| \leq \delta_n s\sqrt{2n} = \beta$ with overwhelming probability. □

D Round Complexity for Malicious Users

We provide here the full proof of Theorem 2. Since the mask $\mathbf{y}$ here is adversarially chosen, we cannot directly rely on the usual rejection sampling lemmas as Lemma 5, to estimate the rejection probability. This is why we need the following lemma, which estimates the rejection probability for an adversarially chosen mask $\mathbf{y}$ and bit b, and an honest center $\boldsymbol{\sigma}_0$ sampled from a discrete gaussian distribution.

Lemma 5. *Let Λ be a full rank lattice, $V \subseteq R^2$ be a set of polynomials of norm at most β and $\rho : V \times \{0,1\} \to [0,1]$ be a probability distribution. Fix the standard deviation s_0 such that $s = \gamma\delta_n s_0\sqrt{2n}$, and $\beta = 2\gamma\delta_n^2 s_0 n$, where $\gamma > 0$. We consider the following algorithm:*

– *Sample* $(\mathbf{y}, b) \hookleftarrow\$ \rho$ *and* $\mathbf{v} \hookleftarrow\$ D^2_{\Lambda,s_0}$*, set* $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{v}$*, and return* $(\mathbf{z}, \mathbf{v}) \leftarrow \mathsf{Reject}(\mathbf{z}, \mathbf{v}, s)$ *as defined in Figure 2.*

Then, the probability that $(\mathbf{z}, \mathbf{v}) \neq \perp$ *is at least* $1/\tilde{M}$ *for*

$$\tilde{M} := \exp\left(\frac{1+\delta_n^2}{2\delta_n^2\gamma^2} + \frac{\sqrt{2\pi}}{\gamma}\right).$$

Proof. Let $M = \exp(1/(2\gamma^2))$. Recall that $\mathsf{Reject}(\mathbf{z}, \mathbf{v}, s)$ outputs $\perp$ whenever

$$u > f(\mathbf{y}, \mathbf{v}) := \frac{\exp\left(\frac{\|\mathbf{v}\|^2}{2s^2}\right)}{M\cosh\left(\frac{\langle \mathbf{z}, \mathbf{v}\rangle}{s^2}\right)},$$

for $u \hookleftarrow\$ [0, 1)$. To upper bound the probability that $\mathsf{Reject}(\mathbf{z}, \mathbf{v}, s)$ outputs $\perp$, we need to lower bound the expected value of $f(\mathbf{y}, \mathbf{v})$ for a given $(\mathbf{y}, b) \in V \times \{0, 1\}$ and $\mathbf{v} \hookleftarrow\$ D^2_{\Lambda,s_0}$. First, we know that $\forall x \in \mathbb{R}$, $1/\cosh(x) \geq e^{-|x|}$, hence we can rewrite

$$f(\mathbf{y}, \mathbf{v}) \geq \frac{1}{M}\exp\left(\frac{1}{s^2}\left(\frac{1}{2}\|\mathbf{v}\|^2 - |\langle \mathbf{z}, \mathbf{v}\rangle|\right)\right).$$

Since $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{v}$, we have

$$f(\mathbf{y}, \mathbf{v}) \geq \frac{1}{M}\exp\left(\frac{1}{s^2}\left(-\frac{1}{2}\|\mathbf{v}\|^2 - |\langle \mathbf{y}, \mathbf{v}\rangle|\right)\right).$$

For the first part, we know that $\mathbb{E}_{\mathbf{v}\hookleftarrow\$D^2_{\Lambda,s_0}}[\|\mathbf{v}\|^2] \leq 2ns_0^2$ from [38, Lemma 4.3], such that we have

$$\mathbb{E}_{\mathbf{v}\hookleftarrow\$D^2_{\Lambda,s_0}}\big[-\|\mathbf{v}\|^2/(2s^2)\big] \geq -1/(2\delta_n^2\gamma^2). \tag{10}$$

As for the second part, since $\|\mathbf{y}\| \leq \beta$, from [34, Lemma 4.3] we have the bound

$$\Pr[|\langle \mathbf{y}, \mathbf{v}\rangle| > r \colon \mathbf{v} \hookleftarrow\$ D^2_{s_0}] \leq 2\exp\left(\frac{-r^2}{2\beta^2 s_0^2}\right).$$

It results that

$$\begin{aligned}\mathbb{E}_{\mathbf{v}\hookleftarrow\$D^2_{\Lambda,s_0}}[|\langle \mathbf{y}, \mathbf{v}\rangle|] &= \int_0^\infty \Pr[|\langle \mathbf{y}, \mathbf{v}\rangle| > t \colon \mathbf{v} \hookleftarrow\$ D^2_{s_0}]dt \\ &\leq 2\int_0^\infty \exp\left(\frac{-t^2}{2\beta^2 s_0^2}\right)dt \\ &\leq 2s_0\beta\sqrt{\frac{\pi}{2}}.\end{aligned}$$

We obtain that

$$\mathbb{E}_{\mathbf{v}\hookleftarrow\$D^2_{\Lambda,s_0}}\big[-|\langle \mathbf{y}, \mathbf{v}\rangle|/s^2\big] \geq -\sqrt{2\pi}/\gamma. \tag{11}$$

Finally by putting (10) and (11) together, from Jensen's inequality, we obtain the lower bound

$$\mathbb{E}_{\mathbf{v}\leftarrow\$D^2_{\Lambda,s_0}}[f(\mathbf{y},\mathbf{v})] \geq \frac{1}{M}\exp\left(-\frac{1}{2\delta_n^2\gamma^2}-\frac{\sqrt{2\pi}}{\gamma}\right).$$

This concludes the proof, by replacing $M = \exp(1/(2\gamma^2))$ in

$$\frac{1}{\tilde{M}} := \frac{1}{M}\exp\left(-\frac{1}{2\delta_n^2\gamma^2}-\frac{\sqrt{2\pi}}{\gamma}\right).$$

□

With this lemma, we are finally ready to prove Theorem 2.

Proof (Theorem 2). We compute $(\mathsf{vk},\mathsf{sk})$ with $(\mathsf{pk}_\mathsf{E},\mathsf{sk}_\mathsf{E}) \leftarrow \mathsf{E.KeyGen}(\lambda)$ generated with an associated secret key, the adversary cannot distinguish this from an honest setup. Let rq be the request sent by $\mathcal{A}$ to start a new round, respectively $\mathsf{rq} = (h, c, \pi, c_\mathsf{nc}, c_m)$ in the first round or $\mathsf{rq} = (h, c, \pi, c_\mathsf{nc}, \mathsf{pnc})$ in the following rounds. Either way, we decrypt the ciphertext c into $(b, \mathbf{y}, \mathbf{r})$. Let $\mathsf{psig} = (\boldsymbol{\sigma}_0, \mathsf{nc}_\mathsf{S})$ be the signer's answer. If the adversary wants to start a new round, $\mathcal{A}$ will have to reveal nc such that $c_\mathsf{nc} = F(\mathsf{nc})$, and for $u = F_0(\mathsf{nc}, \mathsf{nc}_\mathsf{S})$, prove that

$$\mathsf{Reject}(\mathbf{y} + (-1)^b\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_0, s; u) = \perp.$$

From Lemma 5, since u is the output of the random oracle F_0 on fresh randomness nc_S, hence u is uniformly distributed, the statement is true with probability at least $1/\tilde{M}$. If the statement is false, the adversary can still succeed in the two following ways:

- $\mathcal{A}$ found a hash collision $F(\mathsf{nc}) = F(\mathsf{nc}')$,
- $\mathcal{A}$ proved a false statement.

Let ε_F be the advantage of $\mathcal{A}$ against the collision resistance of F. It follows that the probability that $\mathcal{A}$ successfully starts a new round is at most

$$\left(1-\frac{1}{\tilde{M}}\right) + \varepsilon_F + \mathsf{Adv}^{\Pi}_{\mathsf{SND}}(\mathcal{A}).$$

□

E MSIS Hardness

We estimate the hardness of the instance MSIS_B for $B = 2\beta + \beta_0$, for our two parameter selections. We compute the smallest BKZ block size b that can output vectors of norm less than B. Assuming the Gaussian heuristic and the geometric series assumption, BKZ-b outputs vectors of size [14]

$$\|b_1\| \leq \delta_b^{d-1}\det(\mathcal{L})^{1/d},$$

for d the dimension of the lattice $\mathcal{L} = \Lambda_q^\perp(A)$, $\det(\mathcal{L})$ its determinant and δ_b the root Hermite factor [14]. We compute the optimal subdimension d as in [2], as

$$d = \left\lceil \frac{2n \log q}{\log B} \right\rceil,$$

and we know that $\det(\mathcal{L}) = q^n$. The root Hermite factor only depends on the block size b and is defined by

$$\delta_b = \left(\frac{b}{2\pi e} (\pi b)^{\frac{1}{b}} \right)^{1/2(b-1)}.$$

Thus, by iterating over the block sizes b until $B \leq \delta_b^{d-1} \det(\mathcal{L})^{1/d}$, we find that $b = 398$ (resp. 484) is the minimal block size to solve the MSIS_B instance for $n = 1296$ (resp. $n = 1536$). Since the classical cost of BKZ-b is of 0.292 b operations, we obtain a security of 116 bits (resp. 141 bits).

We also detail our parameter selection for our choices of instantiations in Table 3.

Param.	**Instantiation**			**Description**
n	1296		1536	dimension of the ring R
m	3888		4608	ring conductor
λ	116	121	148	security parameter
q	1018523809 $\approx 2^{29}$	535260961 $\approx 2^{28}$	1013921281 $\approx 2^{29}$	prime modulus
γ	2.9	2.1	2.5	rejection parameter
M	1.1	1.1	1.1	$\mathbb{E}$(#rounds) for honest users
$\tilde{M}$	≤ 2.6	≤ 3.8	≤ 3.0	$\mathbb{E}$(#rounds) for malicious users
δ	< 1.0040	< 1.0039	< 1.0034	root Hermite factor
δ_n	1.68	1.68	1.66	Gaussian tail-cut parameter
s_0	$1.5\sqrt{q}$			standard deviation of $\boldsymbol{\sigma}_0$
β_{s_0}	$\delta_n s_0 \sqrt{2n}$			bound for a valid $\boldsymbol{\sigma}_0$
s	$\gamma\beta_0$			standard deviation of $\mathbf{y}$
β	$\delta_n s \sqrt{2n}$			bound for a valid $\boldsymbol{\sigma}$
$\lvert\mathsf{tr}\rvert$	110 KB			transcript size
$\lvert\mathsf{sig}\rvert$	4.70 KB	4.54 KB	5.57 KB	blind signature size

Table 3: A parameter selection for our scheme. We propose two different sets of parameters, for ring dimensions $n = 1296 = 2^4 3^4$ and $n = 1536 = 2^9 3^1$. The prime modulus q is selected such that $q \equiv 1 \mod m$.