Practical, Round-Optimal Lattice-Based Blind Signatures

Shweta Agrawal*, Elena Kirshanova**, Damien Stehlé***, and Anshu Yadav†

Abstract. Blind signatures have numerous applications in privacy-preserving technologies. While there exist many practical blind signatures from number-theoretic assumptions, the situation is far less satisfactory from post-quantum assumptions. In this work, we provide the first round-optimal, practical lattice based blind signature.

Our scheme relies on the Gentry, Peikert and Vainkuntanathan signature [STOC'08] and non-interactive zero-knowledge proofs for linear relations with small unknowns, which are significantly more efficient than their general purpose counterparts. Its security stems from a new and arguably natural assumption which we introduce: one-more-ISIS. This assumption can be seen as a lattice analogue of the one-more-RSA assumption by Bellare *et al* [JoC'03]. To gain confidence, we provide a detailed overview of diverse attack strategies.

1 Introduction

Blind signatures are a fundamental cryptographic primitive with numerous applications in e-cash [23], e-voting [42] cryptocurrencies [66] and other privacy-preserving technologies [67]. In a blind signature scheme [23], a user $\mathcal{U}$, holding a public key and message, may request a signature from a signer $\mathcal{S}$, holding a signing key, such that the signer is not able to link a message-signature pair with a protocol execution, and the user is not able to forge signatures even after multiple interactions with the signer.

Blind signatures have been studied for several decades, and admit many instantiations from a variety of assumptions [24, 59, 33, 44, 34, 35, 31, 49]. Given their wide applicability, there has been a significant thrust towards obtaining practical efficiency. Constructions based on standard assumptions are primarily feasibility results [35, 31], so, in the number-theoretic regime, reasonable new assumptions have been introduced to obtain efficient constructions. For instance, in the group setting, several candidates [24, 55, 59, 40, 33] are based on the hardness of the non-standard ROS/mROS problem[1] or rely [1, 64] on the algebraic group model [44] and the generic group model [54] which are strong idealizations, often implying non-standard assumptions. The situation is analogous in the regime of pairings [19, 17, 34] or RSA [13].

Post-Quantum Regime. In the post-quantum regime, the situation is much more unsatisfactory, especially in the context of optimal round complexity. Lattice-based blind signatures have been provided [61, 6, 5, 21, 46, 56] but were subsequently found to have errors in the security proofs [41]. The recent construction by Hauck *et al.* [41] aimed to fix the errors but the resulting construction is completely impractical – using their suggested parameters, the constructed blind signature has size ≈ 7.73MB, for security against adversaries limited to getting 7 signatures. The very recent work of Lyubashevsky *et al.* [49] achieves better parameters, but their the cost of the signing algorithm grows linearly on the maximum number of signatures that an adversary can query.

* IIT Madras, shweta.a@cse.iitm.ac.in
** ENS de Lyon, elenakirshanova@gmail.com
*** ENS de Lyon and Institut Universitaire de France, damien.stehle@ens-lyon.fr
† IIT Madras, anshu.yadav06@gmail.com

[1] In fact, the ROS problem was recently broken [15].

Our Results. In this work, we provide the first overall practical lattice based blind signatures, with optimal round complexity. Our scheme relies on the Gentry, Peikert and Vainkuntanathan signature [36] and non-interactive zero-knowledge proofs for linear relations with small unknowns, which are significantly more efficient than their general purpose counterparts. Its security stems from a new and arguably natural assumption which we introduce: one-more-ISIS. This assumption can be seen as a lattice analogue of the one-more-RSA assumption by Bellare *et al.* [JoC'03]. Informally, the one-more-ISIS assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell + 1$ GPV signatures [36], even when given access to up to ℓ inversions of arbitrary syndromes.

This construction supports an unbounded number of signatures and achieves overall superior computational cost than all prior candidates. On the other hand, it is based on a new assumption. We believe that for a practice oriented primitive like blind signatures, it is justified to introduce new, plausible assumptions as was done in the number-theoretic regime (as discussed above). We provide detailed cryptanalysis attempts to justify our new assumption.

Table 1 provides a comparison between the round optimal approaches. The contents of the table are only rough estimates, detailed in the corresponding sections.

Construction	**Sig Size**	**User Time**	**Signer Time**	**Transcript Size**	**Verifier Time**
SIS [49]	$\sim$ 150 kB	Depends on maximum number of signatures	Not given	16 MB	Depends on maximum number of allowed signatures
Fischlin, Sec. 3	$\sim$ 130 kB	up to 1h	$<$ 1 ms	$\sim$ 5 kB	Few secs
one-more-ISIS, Sec 4	$\sim 30 - 100$ kB	$<$ 100 ms	$<$ 1 ms	$<$ 2 kB	Few ms

Table 1 Comparison of two-round blind-signature schemes.

1.1 Our Techniques

As a starting point, we adapt Fischlin's scheme [31] to the ROM and instantiate it with efficient lattice based signatures and NIZKs – please see Section 3 for details. Due to the extensive research in efficient lattice based signatures [36, 48, 30, 38, 32, 11, 28] and proof systems [47, 26, 14, 65, 20, 30, 29, 50] over the last 15 years, this already provides a candidate which is "somewhat reasonable" in practice. In our lattice adaptation using GPV "Hash and Sign" signatures [36], the final signature in the blind signature protocol is a NIZK argument of knowledge (NIZKAoK) that the user knows a GPV signature for an *encryption* of the message. In more detail, the user must provide a NIZKAoK for the following statement: Given $(\mathbf{C}, \mathsf{PKE.pk})$ and μ, there exists r and a vector $\mathbf{y}$ such that

$$\|\mathbf{y}\| \leq \beta \ \wedge \ \mathbf{C}\mathbf{y} = H(\mathsf{Enc}(\mathsf{PKE.pk}, \mu; r)).$$

Since the witness r is inside a ciphertext, which in turn is inside a hash function, the statement becomes very complex and using state of art general purpose NIZKAoK [14, 9], we estimate a proof size of more than 100kB and prover time complexity of possibly one hour or more. In the blind signature application, the proof is the signature and the prover is the user. This is very dissatisfying because signature size and user time complexity are often the most important parameters in a blind signature.

The starting point of our protocol based on the one-more-ISIS assumption is to observe that the primary source of inefficiency in Fischlin's protocol is the use of general purpose NIZKs. Our main new idea is to leverage a new, arguably natural assumption, which we call one-more-ISIS so that the problematic general purpose NIZKAoK above may be replaced by an efficient lattice based proof for linear statements with small coefficients. As discussed above, there are now several practical constructions for such statements with proofs

in the tens of kB, and very efficient prover and verifier times. By virtue of our new assumption, the user now needs to prove the following statement: Given $(\mathbf{C}, \mathsf{PKE.pk})$, ct and μ, there exists r and vector $\mathbf{y}$ such that

$$\|\mathbf{y}\| \leq 2\beta \wedge \ \mathbf{C}\mathbf{y} = H(\mu) \wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{y}; r).$$

The above statement also involves the hash function H modeled as a random oracle in the security proof. But the input μ to H is known, implying that $H(\mu)$ can be seen as a public quantity. By using Regev's encryption scheme [60] (or variants of it), one sees that the statement to be proved is linear in the unknowns, which are themselves required to be small.

The one-more-ISIS *Assumption.* Next, we state our assumption and argue about its plausibility. The one-more-$\mathsf{ISIS}_{q,n,m,\sigma,\beta}$ assumption is defined using the following experiment between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$. First, $\mathcal{C}$ uniformly samples a matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and sends it to $\mathcal{A}$. Then $\mathcal{A}$ adaptively makes two types of queries: syndrome queries, to which $\mathcal{C}$ replies with a uniformly sampled vector $\mathbf{t} \leftarrow \mathbb{Z}_q^n$, and preimage queries, where $\mathcal{A}$ queries a vector $\mathbf{t}' \in \mathbb{Z}_q^n$, to which $\mathcal{C}$ replies with a short vector $\mathbf{y}' \leftarrow D_{\mathbb{Z}^m,\sigma}$ such that $\mathbf{C}\mathbf{y}' = \mathbf{t}'$. If ℓ is the total number of preimage queries, we ask the adversary to output $\ell + 1$ pairs of the form $\{(\mathbf{y}_j, \mathbf{t}_j)\}_{j \in [\ell+1]}$, such that $\mathbf{C}\mathbf{y}_j = \mathbf{t}_j$, $\|\mathbf{y}_j\| \leq \beta$ and $\mathbf{t}_j$ were provided via syndrome queries, for all $j \in [\ell + 1]$. We say that the one-more-$\mathsf{ISIS}_{q,n,m,\sigma,\beta}$ problem is hard if the probability that $\mathcal{A}$ succeeds in the above game is negligible.

Note that this definition is reminiscent to the chosen target version of the one-more-RSA inversion problem from [13]. It is also closely related to the k-SIS problem [18] which was introduced in the context of linearly homomorphic signatures. The k-SIS problem is as follows: Given a matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$, and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \mathbb{Z}^m$ satisfying $\mathbf{A} \cdot \mathbf{e}_i = \mathbf{0} \bmod q$, find a short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{A} \cdot \mathbf{e} = \mathbf{0} \mod q$, such that $\mathbf{e}$ is not in $\mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$. In [18], the linearly homomorphic signature must intuitively sign a subspace. Hence for k-SIS, the goal is to restrict the attacker to the subspace of the signatures she has already seen; this prevents her from obtaining signatures of vectors out of the vector subspace that has already been signed. In contrast, in our one-more-ISIS, we do not want to restrict the subspace and indeed allow the attacker to query the oracle more times than the dimension of the whole space. But we are more demanding on the norm of the vector that the attacker must find. We are optimistic that this assumption may have other applications.

To justify our assumption, we attempted to cryptanalyze it. For some parameter regimes, the problem can be solved in polynomial time but, as far as we know, the problem is exponentially hard for the regimes that we use in the blind signature scheme. Broadly, we consider two approaches to solve one-more-ISIS: combinatorial and lattice-based algorithms, and we provide complexity results for one-more-ISIS using these approaches. We also formulate new cryptanalytic questions that the one-more-ISIS assumption raises.

Other Related Works. Aside from lattice based blind signatures, there are a few other constructions from post-quantum assumptions. The most relevant to our work is the code-based construction of Blazy *et al.* [16], relying on the CFS signature scheme [25] and Stern zero-knowledge proofs [63]. Like in our one-more-ISIS construction, their construction relies on a new assumption, related to CFS. However, there are important differences with our work. In CFS, not all syndromes can be inverted, and the procedure needs to be repeated if no inversion is possible. Hence, the resulting blind signature scheme is not round optimal. Moreover, due to the poor scaling of CFS signatures and the use of Stern proofs, their construction achieves signature size of several MB. A blind signature based on multivariate polynomial systems was described in [58], with a non-standard unforgeability security property.

2 Preliminaries

In this section, we provide some preliminaries used in our work.

Notation. We write vectors with bold small letters and matrices with bold capital letters. For any vector $\mathbf{v}$, we denote its ith element by $\mathbf{v}[i]$ or $\mathbf{v}_i$. Similarly, for any matrix $\mathbf{M}$, $\mathbf{M}[i][j]$ or $\mathbf{M}_{ij}$ represents the element in the jth column of ith row. Let S be any set, then $|S|$ represents the cardinality of S, while in case of

any $x \in \mathbb{R}$, $|x|$ represents absolute value of x. For any $n \in \mathbb{N}$, we let the set $\{1, 2, \ldots, n\}$ be denoted by $[n]$.For a distribution D over a countable set $\mathcal{X}$, we let $H_\infty(D) = -\max_{x \in \mathcal{X}} \log_2 D(x)$ denote the min-entropy of D. The statistical distance between two distributions D_0 and D_1 over $\mathcal{X}$ is defined as $\frac{1}{2}\sum_{x \in \mathcal{X}} |D_0(x) - D_1(x)|$.

We use standard definitions for pseudo-random functions (PRF), public-key encryption (PKE) and signatures.

We place ourselves in a setup that allows the attackers to run in time $2^{o(\lambda)}$ and succeed with probability $2^{-o(\lambda)}$, but that forbids them to make more than $\text{poly}(\lambda)$ interactions with honest users. Compared to the setup of polynomially bounded attackers, this allows to better reflect practice and to better differentiate between operations that the adversary can do on its own and are only limited by the adversary runtime (such as hash queries) and operations that require interaction with a honest user and are much more limited (such as signature queries). We note that if we limit ourselves to polynomially bounded adversaries, then all our reductions of our security proofs involve polynomial-time reductions and would not require subexponential hardness assumptions.

2.1 Blind Signatures

To begin, we introduce some notation for interactive executions between algorithms $\mathcal{X}$ and $\mathcal{Y}$. By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$, we denote the joint execution of $\mathcal{X}$ and $\mathcal{Y}$ where $\mathcal{X}$ has private input x, $\mathcal{Y}$ has private input y and $\mathcal{X}$ receives private output a while $\mathcal{Y}$ receives private output b.

Definition 2.1 (Blind Signature). *A blind signature scheme BS consists of PPT algorithms Gen, Vrfy along with interactive PPT algorithms $\mathcal{S}$, $\mathcal{U}$ such that for any λ:*

- $\mathsf{Gen}(1^\lambda)$ *generates a key pair* $(\mathsf{BSig.sk}, \mathsf{BSig.vk})$.
- *The joint execution of* $\mathcal{S}(\mathsf{BSig.sk})$ *and* $\mathcal{U}(\mathsf{BSig.vk}, \mu)$*, where* $\mu \in \{0,1\}^*$*, generates an output* σ *for the user and no output for the signer; this is denoted as* $(\perp, \sigma) \leftarrow \langle \mathcal{S}(\mathsf{BSig.sk}), \mathcal{U}(\mathsf{BSig.vk}, \mu) \rangle$.
- *Algorithm* $\mathsf{Vrfy}(\mathsf{BSig.vk}, \mu, \sigma)$ *outputs a bit* b.

The scheme must satisfy completeness: for any $(\mathsf{BSig.sk}, \mathsf{BSig.vk}) \leftarrow \mathsf{Gen}(1^\lambda)$*,* $\mu \in \{0,1\}^*$ *and* σ *output by* $\mathcal{U}$ *in the joint execution of* $\mathcal{S}(\mathsf{BSig.sk})$ *and* $\mathcal{U}(\mathsf{BSig.vk}, \mu)$*, it holds that* $\mathsf{Vrfy}(\mathsf{BSig.vk}, \mu, \sigma) = 1$ *with probability* $1 - \lambda^{-\omega(1)}$.

Blind signatures must satisfy two security properties: one more unforgeability and blindness [43].

Definition 2.2 (One More Unforgeability). *The blind signature* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *is one more unforgeable if for any polynomial* Q_S*, and any algorithm* $\mathcal{U}^*$ *with run-time* $2^{o(\lambda)}$*, the success probability of* $\mathcal{U}^*$ *in the following game is* $2^{-\Omega(\lambda)}$*:*

1. $\mathsf{Gen}(1^\lambda)$ *outputs* $(\mathsf{ssk}, \mathsf{svk})$*, and* $\mathcal{U}^*$ *is given* svk.
2. *Algorithm* $\mathcal{U}^*$ *interacts concurrently with* Q_S *instances* $\mathcal{S}^1_{\mathsf{ssk}}, \ldots, \mathcal{S}^{Q_S}_{\mathsf{ssk}}$.
3. *Algorithm* $\mathcal{U}^*$ *outputs* $(\mu_1, \sigma_1, \ldots, \mu_{Q_S+1}, \sigma_{Q_S+1})$.

Algorithm $\mathcal{U}^*$ *succeeds if the* μ_i*'s are distinct and* $\mathsf{Vrfy}(\mathsf{svk}, \mu_i, \sigma_i) = 1$ *for all* $i \in [Q_S + 1]$.

The blindness condition says that it should be infeasible for any malicious signer $\mathcal{S}^*$ to decide which of two messages μ_0 and μ_1 of its choice has been signed first in two executions with a honest user $\mathcal{U}$. If one of these executions has returned $\perp$, then the signer is not informed about the other signature either. We will focus on the following notion of honest signer blindness.

Definition 2.3 (Honest Signer Blindness). *The blind signature* $\mathsf{BS} = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *satisfies honest signer blindness if for any algorithm* $\mathcal{S}^*$ *with run-time* $2^{o(\lambda)}$*, the advantage of* $\mathcal{S}^*$ *in the following game is* $2^{-\Omega(\lambda)}$*:*

1. $\mathsf{Gen}(1^\lambda)$ *outputs* $(\mathsf{ssk}, \mathsf{svk})$ *and gives it to* $\mathcal{S}^*$*; algorithm* S^* *outputs two messages* μ_0, μ_1 *of its choice.*

2. *A random bit b is chosen and $\mathcal{S}^*$ interacts concurrently with $\mathcal{U}_0 := \mathcal{U}(\mathsf{svk}, \mu_b)$ and $\mathcal{U}_1 := \mathcal{U}(\mathsf{svk}, \mu_{\bar{b}})$ possibly maliciously; when $\mathcal{U}_0$ and $\mathcal{U}_1$ have completed their executions, the values $\sigma_b, \sigma_{\bar{b}}$ are defined as follows:*
   - *If either $\mathcal{U}_0$ or $\mathcal{U}_1$ aborts, then $(\sigma_b, \sigma_{\bar{b}}) := (\perp, \perp)$.*
   - *Otherwise, let σ_b (resp. $\sigma_{\bar{b}}$) be the output of $\mathcal{U}_0$ (resp. $\mathcal{U}_1$).*

   Algorithm $\mathcal{S}^*$ is given (σ_0, σ_1).
3. *Algorithm $\mathcal{S}^*$ outputs a bit b'.*

Algorithm $\mathcal{S}^*$ succeeds if $b' = b$. If succ *denotes the latter event, then the advantage of $\mathcal{S}^*$ is defined as* $|\Pr[\mathsf{succ}] - 1/2|$.

Full-fledged blindness lets the adversary $\mathcal{S}^*$ sample its own pair $(\mathsf{ssk}, \mathsf{svk})$ at Step 1 (possibly maliciously), and gives svk to the challenger.

2.2 Non-Interactive Zero Knowledge Arguments

Definition 2.4 (Non Interactive Zero Knowledge Argument). *A non-interactive zero-knowledge (*NIZK*) argument system Π for an NP relation R consists of three PPT algorithms* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ *with the following syntax:*

- $\mathsf{Gen}(1^\lambda) \to \mathsf{crs}$: *On input a security parameter λ, the* Gen *algorithm outputs a common reference string* crs*; in the random oracle model, this algorithm may be skipped, since the* crs *can be generated by* P *and* V *by querying the random oracle on some fixed value.*
- $\mathsf{P}(\mathsf{crs}, x, w) \to \pi$: *On input the common reference string* crs*, a statement $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$, a witness w such that $(x, w) \in R$, the prover* P *outputs a proof π.*
- $\mathsf{V}(\mathsf{crs}, x, \pi) \to \mathsf{accept}/\mathsf{reject}$: *On input a common reference string* crs*, a statement $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$ and a proof π, the verifier* V *outputs* accept *or* reject.

The argument system Π should satisfy the following properties.

- ***Completeness:*** *For any $(x, w) \in R$, we have*

$$\Pr[\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda), \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w) : \mathsf{V}(\mathsf{crs}, x, \pi) = 1] \geq 1 - \lambda^{-\omega(1)}.$$

- ***Soundness:*** *For any $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$ and any $2^{o(\lambda)}$ time prover P^*, we have*

$$\Pr[\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda), \pi \leftarrow \mathsf{P}^*(\mathsf{crs}, x) : \mathsf{V}(\mathsf{crs}, x, \pi) = 1] \leq 2^{-\Omega(\lambda)}.$$

- ***Honest Verifier Zero Knowledge:*** *There is a PPT simulator* Sim *such that, for all statements x for which there exists w with $R(x, w) = 1$, for any $2^{o(\lambda)}$ time adversary $\mathcal{A}$, we have:*

$$\begin{aligned} &\Big| \Pr\left[1 \leftarrow \mathcal{A}\left((\mathsf{crs}, x, \pi) \ : \ \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda),\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\right)\right] \\ &- \Pr\left[1 \leftarrow \mathcal{A}\left((\mathsf{crs}, x, \pi) \ : \ (\mathsf{crs}, \pi) \leftarrow \mathsf{Sim}(1^\lambda, x)\right)\right] \Big| \leq 2^{-\Omega(\lambda)}. \end{aligned}$$

Definition 2.5 (Argument of Knowledge). *The argument system* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ *is called an argument of knowledge for the relation R if it is complete and knowledge-sound as defined below.*

- ***Knowledge Sound:*** *For any $2^{o(\lambda)}$ time prover P^*, there exists an extractor $\mathcal{E}$ with expected run-time polynomial in λ and the run-time of P^*, such that for all PPT adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda), (x, s) \leftarrow \mathcal{A}(\mathsf{crs}), \\ \pi^* \leftarrow \mathsf{P}^*(\mathsf{crs}, x, s), b \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi^*), \\ w \leftarrow \mathcal{E}^{\mathsf{P}^*(\mathsf{crs}, x, s)}(\mathsf{crs}, x, \pi^*, b) \end{array} \middle| (x, w) \notin R \wedge b = \mathsf{accept} \right] \leq 2^{-\Omega(\lambda)}.$$

If an argument of knowledge is also non-interactive zero knowledge, it is termed as a non-interactive zero knowledge argument of knowledge, abbreviated as $\mathsf{NIZKAoK}$.

2.3 Lattices and Discrete Gaussians

An *m-dimensional integral lattice* Λ is a full-rank subgroup of $\mathbb{Z}^m$. Among these lattices are the "q-ary" lattices defined as follows: for any integer $q \geq 2$ and any $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define

$$\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{0} \bmod q\}.$$

For a vector $\mathbf{u} \in \mathbb{Z}_q^n$, we define the following coset of $\Lambda_q^{\perp}(\mathbf{A})$:

$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}.$$

We have $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{t}$ for any $\mathbf{t}$ such that $\mathbf{A} \cdot \mathbf{t} = \mathbf{u} \bmod q$.

For any vector $\mathbf{c} \in \mathbb{R}^n$ and any real $\sigma > 0$, the (spherical) Gaussian function with standard deviation parameter σ and center $\mathbf{c}$ is defined as:

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\frac{\pi \|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}\right).$$

The Gaussian distribution is $\mathcal{D}_{\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\sigma^n$.

The (spherical) *discrete Gaussian distribution* over a lattice Λ with standard deviation parameter $\sigma > 0$ and center parameter $\mathbf{c}$ is defined as:

$$\forall \mathbf{x} \in \Lambda,\ \mathcal{D}_{\Lambda,\sigma,\mathbf{c}} = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)},$$

where $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. When $\mathbf{c} = \mathbf{0}$, we omit the subscript $\mathbf{c}$.

2.4 Hardness Assumptions

We will need the Learning With Errors (LWE) problem, which is known to be at least as hard as certain standard lattice problems in the worst case [60, 22] when it is appropriately parameterized.

Definition 2.6 (Learning With Errors (LWE)). *Let q, n, m, α be functions of a parameter λ. For a secret $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{q,n,\alpha,\mathbf{s}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by sampling $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and an $e \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^{n+1}$. The Learning With Errors problem $\mathsf{LWE}_{q,n,m,\alpha}$ is as follows: For $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, the goal is to distinguish between the distributions:*

$$D_0(\mathbf{s}) := U(\mathbb{Z}_q^{m \times (n+1)}) \quad \text{and} \quad D_1(\mathbf{s}) := (A_{q,n,\alpha,\mathbf{s}})^m.$$

We say that a $2^{o(\lambda)}$-time algorithm $\mathcal{A}$ solves $\mathsf{LWE}_{q,n,m,\alpha}$ if it distinguishes $D_0(\mathbf{s})$ and $D_1(\mathbf{s})$ with $2^{-\omega(\lambda)}$ advantage (over the random coins of $\mathcal{A}$ and the randomness of the samples), with $2^{-\omega(\lambda)}$ probability over the randomness of $\mathbf{s}$.

Definition 2.7 (Short Integer Solution (SIS)). *Let q, n, m, β be functions of a parameter λ. An instance of the $\mathsf{SIS}_{q,n,m,\beta}$ problem is a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$. A solution to the problem is a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \beta$ and $\mathbf{A} \cdot \mathbf{v} = \mathbf{0} \bmod q$.*

Like LWE, the SIS problem is known to be at least as hard as certain lattice problems in the worst case [2, 52, 36], when it is appropriately parameterized. The same holds for the *inhomogeneous* version of the SIS problem stated below.

Definition 2.8 (Inhomogeneous Short Integer Solution (ISIS)). *Let q, n, m, β be functions of a parameter λ. An instance of the $\mathsf{ISIS}_{q,n,m,\beta}$ problem is a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{t} \leftarrow \mathbb{Z}_q^n$. A solution to the problem is a vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \beta$ and $\mathbf{A} \cdot \mathbf{v} = \mathbf{t} \bmod q$.*

2.5 Lattice Trapdoors

We will use algorithms for generating a random lattice with a trapdoor, and for sampling short vectors in a lattice coset. The first algorithm is derived from [3, 36, 51], whereas the second is derived from [45, 36, 22].

Lemma 2.9. *Let q, n, m be positive integers with $q \geq 2$ and $m \geq 6n \log_2 q$.*

There is a PPT algorithm $\mathsf{TrapGen}(q, n, m)$ that with probability $1 - 2^{-\Omega(n)}$ outputs a pair $(\mathbf{A}, \mathbf{T}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m}$ such that $\mathbf{A}$ is within $2^{-\Omega(n)}$ statistical distance to uniform in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{T}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A})$.

There is a PPT algorithm $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{u}, \sigma)$, which takes as input the above pair $(\mathbf{A}, \mathbf{T})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a sufficiently large $\sigma = \Omega(\sqrt{n \log q \log m})$ and outputs a vector $\mathbf{e}$ from $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}), \sigma}$. Further, with probability $2^{-\Omega(n)}$, we have $\|\mathbf{e}\| \leq \sigma\sqrt{m}$.

We assume that the $\mathsf{SamplePre}$ algorithm provides the same output when invoked with the same input – this can be ensured by generating the randomness used by the algorithm using a PRF (with the given input as argument).

2.6 Other Useful Lemmas

Lemma 2.10 (Leftover Hash Lemma). *Let $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ be a 2-universal hash function family. Then for any random variable $X \in \mathcal{X}$, for $\varepsilon > 0$ such that $\log |\mathcal{Y}| \leq H_\infty(X) - 2\log(1/\varepsilon)$, the distributions*

$$(h, h(X)) \text{ and } (h, \mathcal{U}(\mathcal{Y}))$$

are within statistical distance ε.

Further, the family $\{\mathbf{A} \in \mathbb{Z}_q^{n \times m} : \mathbf{r} \mapsto \mathbf{Ar}\}$ is 2-universal for any prime q.

The following lemma is adapted from [48], which uses a different Gaussian normalization. In our uses of the third item, for simplicity, we will set $k = \sqrt{2/\pi}$, for which the probability upper bound is $\leq 2^{-m}$.

Lemma 2.11 (Adapted from [48, Lemma 4.4]).

1. *For any $k > 0$, $\Pr[|z| > k\sigma; z \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}] \leq 2\exp(-\pi k^2)$.*
2. *For any $\sigma \geq 3$, $H_\infty(\mathcal{D}_{\mathbb{Z}^m,\sigma}) \geq m$.*
3. *For any $k > 1/\sqrt{2\pi}$, $\Pr[\|\mathbf{z}\| > k\sigma\sqrt{2\pi m};\ \mathbf{z} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}] < (k\sqrt{2\pi})^m \exp(\frac{m}{2}(1 - 2\pi k^2))$.*

3 Starting Point: Instantiating Fischlin's Blind Signature

A simple way to obtain a two-round blind signature from lattices is to instantiate Fischlin's construction [31].

3.1 Construction

The construction uses the following building blocks:

1. A hash function $H : \{0,1\}^* \to \mathbb{Z}_q^n$ that will be modeled as random oracle model in the unforgeability proof.
2. A CPA-secure PKE scheme $\mathsf{PKE}(\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ that is perfectly correct.
3. A $\mathsf{NIZKAoK}$ for the statement of Equation (3.1) (see Figure 1).

The construction is provided in Figure 1. The parameters q, n, m, σ are set such that $n = \Omega(\lambda)$, Lemma 2.9 is applicable, and $\mathsf{SIS}_{q,m,n,2\beta}$ is hard with $\beta = \sigma\sqrt{m}$. The completeness of the scheme follows from the choice of β (using the Gaussian tail bound from Lemma 2.11) and the completeness of the $\mathsf{NIZKAoK}$.

Note that Steps 1 and 2 of the signing algorithm can be implemented quite efficiently. Step 3 is much more costly and results in a large signature bit-size. This is because the statement of Equation (3.1) involves the hash function H (in particular, the input of H must be kept secret). Note that we make a non-black-box use of H in the scheme, but require it to be modeled as a random oracle in the unforgeability proof.

Setup. $\mathsf{Gen}(1^\lambda)$: Upon input the security parameter λ, define $n, m, q, \sigma, \beta = 2\sigma\sqrt{m}$ as functions of λ such that $\mathsf{SIS}_{q,n,m,2\beta}$ is hard and the scheme is both efficient and complete; then do the following:

- Run $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and discard $\mathsf{PKE.sk}$.
- Compute $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(n, m, q)$.
- Output $\mathsf{BSig.sk} = \mathbf{T_C}$, $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$.

Signing. $\langle \mathcal{S}(\mathsf{BSig.sk}), \mathcal{U}(\mathsf{BSig.vk}, \mu) \rangle$:

1. **User:** Given the key $\mathsf{BSig.vk}$ and a message μ, user $\mathcal{U}$ does the following:
   - It samples $\mathsf{PKE.Enc}$ randomness r and computes $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mu; r)$.
   - It sends ct to the signer.
2. **Signer:** Upon receiving ct, signer $\mathcal{S}$ does the following:
   - It computes $H(\mathsf{ct})$ and samples $\mathbf{y} \leftarrow \mathsf{SamplePre}(\mathbf{C}, \mathbf{T_C}, H(\mathsf{ct}), \sigma)$; we have that $\mathbf{y}$ is short and $\mathbf{Cy} = H(\mathsf{ct})$.
   - It sends $\mathbf{y}$ to the user.
3. **User:** Upon receiving $\mathbf{y}$, user $\mathcal{U}$ does the following:
   - It verifies that $\mathbf{y}$ is small and $\mathbf{Cy} = H(\mathsf{ct})$ and aborts if this fails.
   - It generates a $\mathsf{NIZKAoK}$ π for following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$ and μ, there exists r and a vector $\mathbf{y}$ such that
   $$\|\mathbf{y}\| \leq \beta \ \wedge \ \mathbf{Cy} = H(\mathsf{Enc}(\mathsf{PKE.pk}, \mu; r)). \tag{3.1}$$
   - The signature is π.

Verifying. The verifier accepts if the proof π is valid, and rejects if it is not.

Fig. 1 Adaptation of Fischlin's Blind Signature.

3.2 Security

We show that the construction satisfies one more unforgeability and blindness.

Theorem 3.1. *Assume that* $\mathsf{SIS}_{q,n,m,2\beta}$ *is hard and the* $\mathsf{NIZKAoK}$ *is knowledge sound. Then the blind signature scheme in Figure 1 is one more unforgeable in the random oracle model.*

Proof. We argue one more unforgeability using the following hybrids.

Hybrid_0: This is the genuine one more unforgeability experiment.

Hybrid_1: In this hybrid, the challenger (which plays the role of the signer) does not discard the decryption key $\mathsf{PKE.sk}$. For every sign query c_j, it uses $\mathsf{PKE.sk}$ to decrypt c_j into a plaintext μ_j (which can be $\perp$ in case decryption fails). It stores the μ_j's.

Hybrid_2: The difference between this hybrid and the previous one is in how the hash and sign queries are answered. On a fresh input c for a hash query, the challenger first samples $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$ and returns $H(c) = \mathbf{Cy}$. To answer a signing query for an input c, the challenger returns the corresponding $\mathbf{y}$ that it must have sampled while answering the hash query for c. If the sign query is made before the corresponding hash query, then the challenger first sets the hash value as above and then returns the corresponding $\mathbf{y}$.

Indistinguishability of hybrids

1. The differences between Hybrid_0 and Hybrid_1 are only concerning the inner computations of the challenger and not its interactions with the adversary. Hence, the two hybrids are identical in the view of the adversary.
2. By Lemma 2.10, the views of the adversary in Hybrid_1 and Hybrid_2 are within statistical distance $(Q_S + Q_H) \cdot 2^{-\Omega(\lambda)}$ from one another, where Q_S is the number of signing queries and Q_H is the number of hash queries[2].

[2] We recall here that $\mathsf{SamplePre}$ is deterministic, without which the claim would not be true.

Assume now that the adversary succeeds in Hybrid_2 with probability ε. When it succeeds, it generates distinct messages $(\mu_i)_{i\leq Q_S+1}$ and corresponding signatures, i.e., proofs $(\pi_i)_{i\leq Q_S+1}$ for the statement of Equation (3.1), such that all these proofs are accepted. As the adversary makes at most Q_S sign queries, at least one of these μ_i's cannot be part of the μ_j's stored by the challenger: let $\mu^\star$ be this message and $\pi^\star$ be the associated proof.

Using the knowledge soundness of the $\mathsf{NIZKAoK}$ on $\pi^\star$, the challenger extracts a witness $(r^\star, \mathbf{y}^\star)$ such that $\|\mathbf{y}^\star\| \leq \beta$ and $\mathbf{C}\mathbf{y}^\star = H(\mathsf{ct}^\star)$ with $\mathsf{ct}^\star = \mathsf{Enc}(\mathsf{PKE.pk}, \mu^\star; r^\star)$. By perfect correctness of PKE, the ciphertext $\mathsf{ct}^\star$ decrypts to $\mu^\star$. By definition, the message $\mu^\star$ cannot have been queried for a signature. However, it must have been queried for a hash, as otherwise the equality $\mathbf{C}\mathbf{y}^\star = H(\mathsf{ct}^\star)$ would hold with probability at most q^{-n}. This implies that the challenger has previously sampled a vector $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$ such that $\mathbf{C}\mathbf{y} = H(\mathsf{ct}^\star)$. By the Gaussian tail bound (Lemma 2.11), we have $\|\mathbf{y}\| \leq \beta = \sigma\sqrt{m}$ and the probability that $\mathbf{y} = \mathbf{y}^\star$ is $2^{-\Omega(\lambda)}$. We conclude that $\mathbf{y} - \mathbf{y}^\star$ is non-zero, has norm $\leq 2\beta$ and satisfies $\mathbf{C}(\mathbf{y} - \mathbf{y}^\star) = \mathbf{0}$, providing a solution to the $\mathsf{SIS}_{q,n,m,2\beta}$ instance $\mathbf{C}$. □

Theorem 3.2. *Assume that* PKE *is* $\mathsf{IND\text{-}CPA}$ *secure and the* $\mathsf{NIZKAoK}$ *is zero-knowledge. Then the blind signature scheme in Figure 1 satisfies honest signer blindness.*

Proof. We argue blindness using the following hybrids.

Hybrid_0: This is the genuine honest signer blindness experiment.
Hybrid_1: In this hybrid, the proofs π_b and $\pi_{\overline{b}}$ are replaced with simulated proofs.
Hybrid_2: In this hybrid, the ciphertexts ct_b and $\mathsf{ct}_{\overline{b}}$ are changed to independent encryptions of 0.

Indistinguishability of hybrids.

1. Hybrid_0 and Hybrid_1 are indistinguishable in the view of the adversary, because of the zero-knowledge property of the $\mathsf{NIZKAoK}$.
2. Hybrid_1 and Hybrid_2 are indistinguishable in the view of the adversary, because of the IND-CPA security of PKE.

In Hybrid_2, the distinguishing advantage of the adversary is 0, because its views for $b = 0$ and $b = 1$ are statistically identical. □

Full-Fledged Blindness. Note that the scheme as stated may not satisfy full-fledged blindness. In particular, if the malicious signer does not discard $\mathsf{PKE.sk}$ in the setup phase, it could use it to decrypt the ciphertexts in the challenge phase and break blindness. However, the security proof above can be extended to handle full-fledged blindness if we can ensure that $\mathsf{PKE.pk}$ has been honestly generated by the adversarial signer, without a corresponding decryption key. For example, if $\mathsf{PKE.pk}$ is computationally indistinguishable from uniform, then we could replace $\mathsf{PKE.pk}$ in the scheme by the output of another hash function H' modeled as a random oracle, on an arbitrary public input. Since the secret key must anyway be discarded in the construction, setting the public key as the output of the random oracle ensures that the adversarial signer cannot know the corresponding secret key. In the (full fledged blindness) security proof, we would then introduce a very first game in which the output of H' is replaced by a properly generated $\mathsf{PKE.pk}$. Note that a maliciously generated $\mathbf{C}$ has no impact on blindness since it is not involved in the user's message to the signer.

3.3 Efficiency Estimate

We consider the following instantiation of the building blocks.

- For PKE, we can take any lattice-based public-key encryption scheme. It is only required to be IND-CPA, but it must be perfectly correct. The latter property can typically be guaranteed by tail-cutting error distributions and increasing the working modulus sufficiently. Also, lattice-based encryption schemes

typically typically have public keys that are computationally indistinguishable from uniform, as required for the full fledged blindness adaptation described above. For example, one could use a variant of the NewHope scheme [7], modified to provide perfect correctness. It is expected that ciphertexts will be of bitlengths below a few kB.

- For the underlying signature scheme, we recommend using the Falcon scheme [32], which is an efficient instantiation of the TrapGen-SamplePre framework from [36]. With this choice, the first transcript $\mathbf{t}$ will have size below 2kB and the second transcript will have size below 1kB. Also, that makes the signer particularly efficient – for instance, using Falcon [32], signing time is in the range $0.15 - 0.3$ ms depending on choice of parameters.
- As the hash function is modeled as a random oracle in the unforgeability proof, one could use SHA-3-256. With the above choices for the public-key encryption and signature schemes, one may need more than 15 rounds for reading the input and a similar number to write the output.
- Unfortunately, as the statement of Equation (3.1) involves a hash function H that is modeled as a random oracle in the unforgeability proof, it seems we are bound to use an all-purpose NIZKAoK. For example, one could use an instantiation of Aurora [14]. Estimating a precise cost is difficult, but we do not expect a proof of size below 100kB. Also, prover complexity could approach 1 hour, whereas verifier runtime could be several seconds. It could be beneficial to use hash functions designed to be compatible with all-purpose NIZKAoK, such as [8, 37].

4 Two Round Blind Signature from One-More-ISIS

In this section, we describe a significantly more practical scheme, under a new assumption.

4.1 The One-More-ISIS Assumption

We first introduce the one-more-ISIS hardness assumption. As it is a new assumption, we provide a detailed assessment of potential attacks, in Subsection 4.5.

Informally, the one-more-ISIS assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell + 1$ GPV signatures [36], even when given access to up to ℓ inversions of arbitrary syndromes. We stress that these are not signature queries, as a query for a message μ corresponds to a *uniformly distributed* syndrome $H(\mu)$ (modelling H by a random oracle), whereas here the attacker is allowed to make inversion queries for *arbitrary* syndromes. As a result, the one-more-ISIS could possibly be easier to solve than it is to break the chosen-message security of the GPV signature scheme.

Definition 4.1. *Let q, n, m, σ, β be functions of security parameter λ. The one-more-ISIS$_{q,n,m,\sigma,\beta}$ assumption is defined using the following experiment.*

1. *The challenger $\mathcal{C}$ uniformly samples a matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and sends $\mathbf{C}$ to $\mathcal{A}$.*
2. *The adversary adaptively makes queries of the following types to the challenger, in any order.*
   - **Syndrome queries.** *The adversary $\mathcal{A}$ requests $\mathcal{C}$ for a challenge vector, to which $\mathcal{C}$ replies with a uniformly sampled vector $\mathbf{t} \leftarrow \mathbb{Z}_q^n$. We denote the set of received vectors by S.*
   - **Preimage queries.** *The adversary $\mathcal{A}$ queries a vector $\mathbf{t}' \in \mathbb{Z}_q^n$, to which $\mathcal{C}$ replies with a short vector $\mathbf{y}' \leftarrow D_{\mathbb{Z}^m,\sigma}$ such that $\mathbf{C}\mathbf{y}' = \mathbf{t}'$. We denote by ℓ the total number of preimage queries.*
3. *In the end, the adversary $\mathcal{A}$ outputs $\ell + 1$ pairs of the form $\{(\mathbf{y}_j, \mathbf{t}_j)\}_{j \in [\ell+1]}$.*
4. *The adversary wins if $\mathbf{C}\mathbf{y}_j = \mathbf{t}_j$, $\|\mathbf{y}_j\| \leq \beta$ and $\mathbf{t}_j \in S$ for all $j \in [\ell + 1]$.*

The one-more-ISIS$_{q,n,m,\sigma,\beta}$ assumption states that for every adversary $\mathcal{A}$ running in time $2^{o(\lambda)}$ making at most $\lambda^{O(1)}$ preimage queries and $2^{o(\lambda)}$ syndrome queries, the probability (over the randomness of $\mathcal{A}$ and $\mathcal{C}$) that $\mathcal{A}$ wins is $2^{-\Omega(\lambda)}$.

The definition is reminiscent to the chosen target version of the one-more-RSA inversion problem from [13]. We could define a variant of one-more-ISIS inspired from the known target version of the one-more-RSA inversion problem from [13], in which the set S is restricted to be of size $\ell + 1$. The choice (chosen target) formulation from Definition 4.1 is driven by the security proof of the scheme. In the RSA setting, the chosen and known target versions reduce to one another, but this seems difficult to adapt to the ISIS setting.

4.2 Construction

The construction uses the following building blocks:

1. A hash function $H : \{0,1\}^* \to \mathbb{Z}_q^n$ that will be modeled as random oracle model in the unforgeability proof.
2. A NIZK for the statement of Equation (4.1) (see Figure 2).
3. A CPA-secure PKE scheme PKE(PKE.KeyGen, PKE.Enc, PKE.Dec) that is perfectly correct.

The construction is provided in Figure 2. The parameters q, n, m, σ are set such that Lemma 2.9 is applicable, the distribution of $\mathbf{Cx}$ is close to uniform at Step 1 of the signing algorithm (using Lemmas 2.9 and 2.10), and one-more-ISIS$_{q,m,n,\sigma,2\beta}$ is hard with $\beta = \sigma\sqrt{m}$.

We include ciphertext $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{y}-\mathbf{x}; r)$ in the signature (see Step 5). It enables to circumvent rewinding in the extraction of all the witnesses $\widetilde{\mathbf{y}}_i = (\mathbf{y}_i - \mathbf{x}_i)$ of the $Q_S + 1$ message-signature pairs output by the adversary, in the proof of unforgeability. Without it, the reduction may need to rewind $Q_S + 1$ times to extract all the $\widetilde{\mathbf{y}}_i$'s, to construct the one-more-ISIS solution, leading to a security loss exponential in Q_S.

Setup. $\mathsf{Gen}(1^\lambda)$: Upon input the security parameter λ, define $n, m, q, \sigma, \beta = \sigma\sqrt{m}$ as functions of λ such that one-more-ISIS$_{q,n,m,\sigma,2\beta}$ is hard and the scheme is both efficient and complete; then do the following:

- Run $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and discard $\mathsf{PKE.sk}$.
- Compute $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(n, m, q)$.
- Output $\mathsf{BSig.sk} = \mathbf{T_C}$, $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$.

Signing. $\langle \mathcal{S}(\mathsf{BSig.sk}), \mathcal{U}(\mathsf{BSig.vk}, \mu) \rangle$:

1. **User:** Given the key $\mathsf{BSig.vk}$ and a message μ, user $\mathcal{U}$ does the following:
   - It samples $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$.
   - It computes $\mathbf{t} = \mathbf{Cx} + H(\mu)$.
   - It sends $\mathbf{t}$ to the signer.
2. **Signer:** Upon receiving $\mathbf{t}$, signer $\mathcal{S}$ does the following:
   - It samples a short vector $\mathbf{y} \leftarrow \mathsf{SamplePre}(\mathbf{C}, \mathbf{T_C}, \mathbf{t}, \sigma)$; we have $\mathbf{Cy} = \mathbf{t}$.
   - It sends $\mathbf{y}$ to the user.
3. **User:** Upon receiving $\mathbf{y}$, user $\mathcal{U}$ does the following:
   - It verifies that $\|\mathbf{y}\| \le \beta$ and $\mathbf{Cy} = \mathbf{t}$ and aborts if this fails.
   - It samples $\mathsf{PKE.Enc}$ randomness r and computes $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{y} - \mathbf{x}; r)$.
   - It generates a NIZK π for following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$, ct and μ, there exists r and a vector $\widetilde{\mathbf{y}}$ such that
   $$\|\tilde{\mathbf{y}}\| \le 2\beta \ \wedge \ \mathbf{C}\tilde{\mathbf{y}} = H(\mu) \ \wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \tilde{\mathbf{y}}; r). \tag{4.1}$$
   - The signature is (π, ct).

Verifying. The verifier accepts if the proof π is valid, and rejects if it is not.

Fig. 2 Blind Signature from one-more-ISIS.

Completeness. We make the following observations to argue completeness. From the correctness of SamplePre, we have $\|\mathbf{y}\| \le \beta$ and $\mathbf{Cy} = \mathbf{t}$, where $\mathbf{t} = \mathbf{Cx} + H(\mu)$. This gives us that $\mathbf{C}(\mathbf{y} - \mathbf{x}) = H(\mu)$. Furthermore, the vector $\mathbf{x}$ satisfies $\|\mathbf{x}\| \le \beta$ by design and hence $\|\mathbf{y} - \mathbf{x}\| \le 2\beta$. Finally, $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{y} - \mathbf{x}; r)$ by construction. Hence, the proof π for Equation (4.1) verifies and the user accepts the proof because of the completeness of NIZKAoK.

4.3 Security

We show that our construction satisfies one more unforgeability and blindness.

Theorem 4.2. *Assume that the* one-more-ISIS$_{q,n,m,\sigma,2\beta}$ *assumption holds and the argument* NIZKAoK *is sound. Then the blind signature scheme in Figure 2 is one more unforgeable in random oracle model.*

Proof. We argue one more unforgeability using the following hybrids.

Hybrid_0: This is the genuine one more unforgeability experiment.

Hybrid_1 : In this hybrid, the challenger does not discard the decryption key PKE.sk. For every signature $\sigma_j = (\pi_j, \mathsf{ct}_j)$ output by the adversary (for $j \in [Q_S + 1]$, where Q_S is the number of signing queries that the adversary issues), it uses PKE.sk to decrypt ct_j into a plaintext $\widetilde{\mathbf{y}}_j$ (which can be $\perp$ in case decryption fails).

Indistinguishability of hybrids. The differences between Hybrid_0 and Hybrid_1 are only concerning the inner computations of the challenger and not its interactions with the adversary. Hence, the two hybrids are identical in the view of the adversary.

Claim. Assume that the NIZK argument system is sound and PKE is correct. If there is an adversary $\mathcal{A}$ that makes at most Q_S signing queries and succeeds in generating $Q_S + 1$ signatures in Hybrid_1, then there exists a one-more-ISIS adversary $\mathcal{B}$ with Q_S preimage queries.

Note that this claim implies the result.

Proof. The reduction is as follows.

1. Upon being challenged by the one-more-ISIS challenger $\mathcal{C}$, with input matrix $\mathbf{C}$, algorithm $\mathcal{B}$ does the following:
   - It samples $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
   - It invokes $\mathcal{A}$ with $(\mathbf{C}, \mathsf{PKE.pk})$ as verification key.
2. In response to each (fresh) hash query on input μ from $\mathcal{A}$, algorithm $\mathcal{B}$ makes a syndrome query to $\mathcal{C}$. Challenger $\mathcal{C}$ returns a uniform vector $\mathbf{t} \in \mathbb{Z}_q^n$, which $\mathcal{B}$ forwards to $\mathcal{A}$ as $H(\mu)$.
3. To answer a signing query on input $\mathbf{t}'$, algorithm $\mathcal{B}$ forwards $\mathbf{t}'$ to $\mathcal{C}$ as a preimage query. Challenger $\mathcal{C}$ returns a short vector $\mathbf{y}'$, such that $\mathbf{C}\mathbf{y}' = \mathbf{t}'$. Algorithm $\mathcal{B}$ forwards $\mathbf{y}'$ to $\mathcal{A}$.
4. Eventually, adversary $\mathcal{A}$ outputs $Q_S + 1$ pairs $\{\mu_j, (\pi_j, \mathsf{ct}_j)\}_{j \in [Q_S+1]}$.
5. If the π_j's pass verification, then algorithm $\mathcal{B}$ decrypts the ct_j's and obtains $Q_S + 1$ corresponding short vectors $\widetilde{\mathbf{y}}_j$. If all μ_j's have been hash-queried by $\mathcal{A}$ and the vectors $\widetilde{\mathbf{y}}_j$ satisfy Equation (4.1) for all $j \in [Q_S + 1]$, then $\mathcal{B}$ outputs $\{(\widetilde{\mathbf{y}}_j, H(\mu_j))\}_{j \in [Q_S+1]}$. If any decryption fails or any of the above conditions are not satisfied, then algorithm $\mathcal{B}$ aborts.

Note that $\mathcal{A}$'s view is identical to the one in Hybrid_1. It hence succeeds with the same probability. We now assume that this is the case. By the perfect correctness of PKE and the soundness of NIZK, the probability that a decryption fails is $\leq (Q_S + 1) \cdot 2^{-\Omega(\lambda)}$. We assume we are not in this situation. We claim that for each μ_j, adversary $\mathcal{A}$ must have issued a corresponding hash query to $\mathcal{B}$. This is because otherwise, there is only a q^{-n} probability that a fresh $H(\mu_j)$ is equal to $\mathbf{C}\widetilde{\mathbf{y}}_j$. Finally, by the soundness of NIZK, the following indeed holds for all $j \in [Q_S + 1]$:

$$\|\widetilde{\mathbf{y}}_j\| \leq 2\beta \ \wedge \ \mathbf{C}\widetilde{\mathbf{y}}_j = H(\mu_j).$$

Next, observe that because of the way hash queries are answered by $\mathcal{B}$, the value $H(\mu_j)$ is one of the syndromes returned by $\mathcal{C}$. Define $\mathbf{t}_j = H(\mu_j)$. Then we get, for all $j \in [Q_S + 1]$,

$$\|\widetilde{\mathbf{y}}_j\| \leq 2\beta \ \wedge \ \mathbf{C}\widetilde{\mathbf{y}}_j = \mathbf{t}_j.$$

Note that $\mathcal{B}$ issues one preimage query for each signing query from $\mathcal{A}$. Since $\mathcal{A}$ can issue at most Q_S signing queries, algorithm $\mathcal{B}$ also issues at most Q_S preimage queries to $\mathcal{C}$. This completes the proof. □

Next we show that the construction satisfies honest signer blindness.

Theorem 4.3. *Assume that* PKE *is* IND-CPA *secure and the* NIZK *is zero-knowledge. Then the blind signature in Figure 2 satisfies honest signer blindness.*

Proof. We argue blindness using following hybrids.

Hybrid_0 : This is the genuine honest signer blindness experiment.
Hybrid_1 : This hybrid differs from the previous one in the way the proofs π_0 and π_1 are computed: instead of computing these honestly, the challenger computes them using the NIZK simulator.
Hybrid_2 : This hybrid differs from the previous hybrid in that both ct_0 and ct_1 encrypt $\mathbf{0}$ instead of $(\mathbf{y}_0 - \mathbf{x}_0)$ and $(\mathbf{y}_1 - \mathbf{x}_1)$, respectively.
Hybrid_3 : This hybrid differs from the previous hybrid in the way the challenger computes $\mathbf{t}_0$ and $\mathbf{t}_1$. Instead of sampling $\mathbf{x}_0$ (resp. $\mathbf{x}_1$) and computing $\mathbf{t}_0 = \mathbf{C}\mathbf{x}_0 + H(\mu_b)$ (resp. $\mathbf{t}_1 = \mathbf{C}\mathbf{x}_1 + H(\mu_{\bar{b}})$), it samples $\mathbf{u}_0$ (resp. $\mathbf{u}_1$) uniformly and sets $\mathbf{t}_0 = \mathbf{u}_0 + H(\mu_b)$ (resp. $\mathbf{t}_1 = \mathbf{u}_1 + H(\mu_{\bar{b}})$).

Indistinguishability of hybrids.

1. The only difference between Hybrid_0 and Hybrid_1 is in the way π_0 and π_1 are computed. The two hybrids are indistinguishable because of the zero-knowledge property of the NIZK.
2. The only difference between Hybrid_1 and Hybrid_2 is in the messages being encrypted by ct_0 and ct_1. The two hybrids are indistinguishable because of the IND-CPA security of PKE.
3. Note that in Hybrid_2, the vectors $\mathbf{x}_0$ and $\mathbf{x}_1$ are only used in the computations of the vectors $\mathbf{t}_0$ and $\mathbf{t}_1$ that the challenger provides to the adversary when it plays the roles of users $\mathcal{U}_0$ and $\mathcal{U}_1$. By the leftover hash lemma (Lemma 2.10) and the fact that $\mathbf{C}$ is statistically close to uniform (Lemma 2.9), we have that $\mathbf{t}_0$ and $\mathbf{t}_1$ are statistically indistinguishable from uniform. Hence, Hybrid_2 and Hybrid_3 are indistinguishable.

Finally, in Hybrid_3, the adversary $\mathcal{S}^*$ has zero advantage in guessing the bit b since, in its view, it is information theoretically hidden. □

Full-Fledged Blindness. The security proof above can be extended to handle full-fledged blindness with the following modifications.

(i) Similarly to the construction in Section 3, by choosing a suitable encryption scheme so that PKE.pk is computationally indistinguishable to uniform, one can set PKE.pk as the output of a random oracle on a publicly-known value. In the blindness security proof, the challenger runs PKE.KeyGen and programs the random oracle to output PKE.pk.
(ii) In addition, we also need to ensure that the matrix $\mathbf{C}$ in BSig.vk is such that $\mathbf{C}\mathbf{x}$ hides $\mathbf{x}$, so that $\mathbf{t}$ is close to uniform. This holds true if the smoothing parameter of $\Lambda_q^{\perp}(\mathbf{C})$ is small, i.e., $\eta_{\varepsilon}(\Lambda^{\perp}(\mathbf{C})) \leq \sigma$ for $\varepsilon \in (0, 1/2)$ (see [36, Lemma 5.2]). To ensure this, we can add a proof that $\mathbf{C}$ has a small smoothing parameter [57]. Note that this proof must be only added once, across all executions of the protocol.

4.4 Efficiency Estimate

From an efficiency perspective, a crucial difference from the scheme provided in Section 3 lies in the specific statement required to be handled by the NIZK (see Figure 2). The statement also involves the hash function H modeled as a random oracle in the security proof. But the input μ to H is known, implying that $H(\mu)$ can be computed publicly and can be seen as a public quantity. By using Regev's encryption scheme [60] (or variants of it), one sees that the statement to be proved is linear in the unknowns, which are themselves required to be small. As a result, we can circumvent the use of a general-purpose NIZK and instead rely on algebraic proofs for linear relations [29, 50]. This lets us reduce the signature size to maybe as small as 50kB [29, 50], against more than 100kB. More importantly, the cost of generating and verifying the proof becomes very small.

Towards a concrete version of the scheme, we suggest instantiating the other building blocks as follows. The hash function could be taken to be SHA-3-256. The encryption scheme could be set as the IND-CPA NewHope [7], properly modified to ensure perfect correctness. The Falcon signature scheme [32] could provide the concrete instantiation of the TrapGen-SamplePre functions. With these choices, the transcripts between the user and the signer are below 2kB, the size of the signature is dominated by the size of the proof, and all algorithms can be run very efficiently (in orders less than a second).

4.5 Security Analysis of One-More-SIS

The purpose of this section is to argue why we believe that the new computational problem we introduce, one-more-ISIS, is hard. We did not succeed in obtaining a reduction from a well-studied problem to one-more-ISIS, but we still expect that for the parameter ranges relevant to our constructions, this problem cannot be solved by polynomial or even sub-exponential time attackers.

The hardness of the one-more-ISIS problem as stated in Definition 4.1 primarily depends on the precise relation between β, the upper bound on the norm of the vectors $\mathbf{y}_i$'s the adversary must output, and the dimensions m and n of the input matrix $\mathbf{C}$. We also assume that σ – the standard deviation parameter of the preimage queries – is of order $\Omega(\sqrt{m})$, which what we would expect from an efficient sampler, e.g. [36]. Note that a significantly smaller standard deviation, e.g., of order $\mathcal{O}(1)$, would invalidate the hardness of the one-more-ISIS assumption as extremely short $\mathbf{y}$'s would enable an adversary to solve one-more-ISIS (see the discussion below). In this section we make the hardness of the one-more-ISIS problem explicit by describing the parameter regimes for which this problem can be solved in polynomial time, and for which, as far as we know, the problem is exponentially hard. We consider two approaches to solve one-more-ISIS: combinatorial attacks and lattice-based attacks.

Combinatorial attacks. We start by showing an elementary polynomial time algorithm that achieves $\beta = \Theta(\sqrt{mn}\sigma)$ and requires $(q \cdot n)$ ISIS preimage oracle calls.

Consider the set of n-dimensional vectors $A = \{\mathbf{e}_i \cdot a : \ i \in [n], a \in \mathbb{Z}_q\}$, where the $\mathbf{e}_i$'s are the canonical-basis vectors. The set A is of size $q \cdot n$. The adversary runs preimage queries for all vectors from A and receives Gaussian vectors $\mathbf{y}'$'s. Thanks to the Gaussian tail bound (see Lemma 2.11), we have $\|\mathbf{y}'\| \leq 2\sqrt{m}\sigma$ with probability greater than $1 - 2^{-m}$ for all $\mathbf{y}'$'s. Any element from $\mathbb{Z}_q^n$, and thus the challenge $\mathbf{t}$, can be expressed as a sum of at most n vectors from A (one for each coordinate). The adversary then sums the corresponding $\mathbf{y}'$'s it received from the ISIS preimage oracle and obtains a new $\mathbf{y}$ such that $\mathbf{C}\mathbf{y} = \mathbf{t}$. The resulting $\mathbf{y}$ is a valid one-more-ISIS solution for $\beta = \Theta(\sqrt{nm} \cdot \sigma)$ with probability greater than $1 - 2^{-\Omega(m)}$.

The algorithm can be generalized to a larger set A. The generalization, presented in Algorithm 3, makes the attack less efficient, but reduces the bound on β. It is parametrized by Q, the upper bound on the number of the preimage queries the attacker can issue. This is also the assumed upper bound on the memory capacity of the attacker, since the attack requires that all the responses are stored.

Input: The ISIS preimage oracle $\mathcal{O}^{\mathsf{ISIS}}(\cdot)$, a number Q of queries to $\mathcal{O}^{\mathsf{ISIS}}$, and $\mathbf{t} \in \mathbb{Z}_q^n$.
Output: A short vector $\mathbf{y} \in \mathbb{Z}_q^m$ such that $\mathbf{C}\mathbf{y} = \mathbf{t} \bmod q$.

1. Set $w = \lfloor \frac{\log(Q/n^2)}{\log q} \rfloor$.
2. Let $A = \left\{ \sum_{w\cdot(i-1)<j\leq \max\{w\cdot i, n\}} \mathbf{e}_j \cdot a_j : \forall i \in \left[\left\lceil \frac{n}{w} \right\rceil \right], a_j \in \mathbb{Z}_q \right\}$.
3. For all $\mathbf{a} \in A$, set $T[\mathbf{a}] = \mathcal{O}^{\mathsf{ISIS}}(\mathbf{a})$.
4. Write $\mathbf{t} = \mathbf{a}_{i_1} + \ldots + \mathbf{a}_{i_{\lceil n/w \rceil}}$.
5. Output $\mathbf{y} = T[\mathbf{a}_{i_1}] + \ldots + T[\mathbf{a}_{i_{\lceil n/w \rceil}}]$.

Fig. 3 Combinatorial Attack on one-more-ISIS.

The correctness of Algorithm 3 is direct: any $\mathbf{t} \in \mathbb{Z}_q^n$ can be efficiently written as a sum of at most $\lceil n/w \rceil$ elements from the set A constructed on Step 2. Note that $|A| \leq n^2 q^w$: by definition of w, the algorithm indeed makes $\leq Q$ queries. Finally, we can bound the norm of the output as $\|\mathbf{y}\| < 2\sqrt{\lfloor \frac{n}{w} \rfloor \cdot m} \cdot \sigma = \Theta(\sqrt{1 + \frac{n \log q}{\log(Q/n^2)}} \cdot \sqrt{m} \cdot \sigma)$, with probability greater than $1 - 2^{-\Omega(m)}$. The algorithm is correct for any $1 \leq w \leq n$ computed on Step 1, providing a trade-off between the runtime (which is essentially the number Q of preimage queries) and the bound on β.

Lattice-based attacks. A strategy to attack one-more-ISIS is to use a discrete Gaussian sampler algorithm [45, 36]. This allows to solve one-more-ISIS in $\mathrm{poly}(m)$ time with $\beta = \Omega(m\sigma)$ using $O(m^2)$ preimage queries. More precisely, the attacker does the following:

1. It queries the preimage ISIS oracle $\Theta(m^2)$ times for $\mathbf{t} = \mathbf{0}$. From the oracle's answers, it computes a basis $\mathbf{B}$ for $\Lambda_q^\perp(\mathbf{C}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{C}\mathbf{y} = \mathbf{0} \bmod q\}$.
2. Given input $\mathbf{t} \in \mathbb{Z}_q^n$, it runs a Gaussian sampler [45, 36] instantiated with the basis $\mathbf{B}$ and the syndrome vector $\mathbf{t}$ (as in Lemma 2.9). It outputs what the sampler replies.

Let us make several remarks about the above procedure. First, thanks to standard properties of lattice Gaussian distributions, it indeed suffices to query the ISIS preimage oracle $\Theta(m^2)$ times in Step 1, in order to obtain a basis of $\Lambda_q^\perp(\mathbf{C})$ with at least constant probability bounded away from 0 (see [60, Corollary 3.16]). Second, the Gaussian sampler from [22] produces samples from any coset of the lattice with standard deviation $\sigma \geq \|\mathbf{B}\|\sqrt{\log m}$, where $\|\mathbf{B}\|$ is the norm of the longest vector in $\mathbf{B}$. Since $\|\mathbf{B}\| \leq 2\sqrt{m} \cdot \sigma$ (with overwhelming probability), the sampler will produce valid one-more-ISIS solutions for $\beta = O(m\sigma\sqrt{\log m})$ in $\mathrm{poly}(m)$ time using $\Theta(m^2)$ calls to the ISIS preimage oracle.

Observing that one-more-ISIS only cares about the norm of the returned $\mathbf{y}$ but not about its actual distribution, we can slightly improve the bound on β by getting rid of the factor $\sqrt{\log m}$. For this purpose, we replace the Gaussian Sampling procedure by Babai's Nearest Plane algorithm [10]. This algorithm receives on input a lattice basis $\mathbf{B} \in \mathbf{Z}^{m \times m}$ and a target vector $\mathbf{z} \in \mathbf{Z}^m$, and outputs a lattice vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{z}\| \leq \frac{1}{2}(\sum_{i \in [m]} \|\mathbf{b}_i\|^2)^{1/2}$. In our case, the right-hand side is bounded from above by $m\sigma$ with probability greater than $1 - 2^{-\Omega(m)}$. We run Babai's Nearest Plane algorithm on input $(\mathbf{B}, \mathbf{z})$, where $\mathbf{z} \in \mathbb{Z}^m$ is an arbitrary vector that satisfies $\mathbf{C}\mathbf{z} = \mathbf{t} \bmod q$. Let $\mathbf{v} = \mathbf{B}\mathbf{c}_\mathbf{v}$ be the output and let $\mathbf{e} = \mathbf{v} - \mathbf{z}$. Then we have $\mathbf{t} = \mathbf{C}\mathbf{z} = \mathbf{C} \cdot \mathbf{B}\mathbf{c}_\mathbf{v} - \mathbf{C}\mathbf{e} = -\mathbf{C}\mathbf{e} \bmod q$ with $\mathbf{e}$ being a valid one-more-ISIS solution for $\beta = \Theta(m\sigma)$.

In Section 4.5, we discussed some approaches for analyzing the one-more-ISIS problem. Can we do better? A strategy to improve the above bounds on β is to obtain basis of the lattice $\Lambda_q^\perp(\mathbf{C})$ that is *shorter* than what the ISIS preimage oracle offers. We can go as far as the Minkowski's bound suggests, i.e., we can achieve $\|\mathbf{B}\| = \lambda_1(\Lambda_q^\perp(\mathbf{C})) \leq \min_{m' \leq m} \sqrt{m'} \cdot q^{n/m'}$ (here we assume that all lattice minima have essentially the same norms, which is expected to be the case when $\mathbf{C}$ is sampled uniformly). The latter bound is $O(\sqrt{n \ln q})$ when $m = \Omega(n \log q)$. Vectors of such a small norm can be found by calling shortest vector problem solvers on $\Lambda_q^\perp(\mathbf{C})$. The fastest known such algorithms run in time $2^{O(m)}$ (see, e.g., [12]). This exponential time attack enables us to solve one-more-ISIS for $\beta = \Theta(\sqrt{mn \ln q})$ by invoking Babai's Nearest Plane algorithm on the obtained short basis. Note that the ISIS preimage oracle is only used to obtain a basis of $\Lambda_q^\perp(\mathbf{C})$. A trade-off between the quality of β and the runtime is possible: a b-BKZ reduction [39, 62] yields a basis $\mathbf{B}$ with $\|\mathbf{B}\| \leq b^{O(m/b)} \cdot \lambda_1(\Lambda_q^\perp(\mathbf{C}))$ in time $2^{\mathcal{O}(b)}$, thus leading to $\beta = b^{O(m/b)} \cdot \sqrt{mn \ln q})$. Note that in order to outperform the bound on β we have in the polynomial time regime, the BKZ parameter b has to be of order $\Theta(m/\log \sigma)$, when $m = \Theta(n \log q)$.

To summarize, we have the run-times for solving one-more-ISIS:

- there exists a combinatorial algorithm that achieves $\beta = \Theta(\sqrt{1 + \frac{n \log q}{\log(Q/n^2)}} \cdot \sqrt{m}\sigma)$ in time Q and using $Q \geq nq$ preimage queries;
- there exists a lattice-based algorithm that achieves $\beta = \Theta(m\sigma)$ in polynomial time using $O(m^2)$ preimage queries; except for very few queries, it is outperformed by the combinatorial algorithm;
- there exists a lattice-based algorithm that achieves $\beta = 2^{O(\frac{m \log\log T}{\log T})}\sqrt{mn \log q})$ in time T without any preimage query (except to obtain a basis of $\Lambda_q^\perp(\mathbf{C})$).

Open questions and potential directions. Let us now formulate some cryptanalytic questions that the new one-more-ISIS hardness assumptions raises.

I. Improving algorithms for the shortest vector problem with preimage queries. One might wonder whether we can accelerate existing shortest vector solvers, such as sieving algorithms [4, 53, 12], once we already have a somewhat short basis. Just from the nature of sieving algorithms it does not seem to be the

case: even to obtain a small constant reduction in the norm of the current shortest vector, sieving generates and processes $2^{\mathcal{O}(m)}$ vectors which already constitutes its asymptotic cost.

II. Improving Babai's Nearest Plane with a short generating set. Given access to ISIS preimages, another direction one can consider is to try to accelerate the *closest vector problem* (CVP) solvers on $\Lambda_q^\perp(\mathbf{C})$, by exploiting the fact that we have many short vectors from this lattice. The presence of many short vectors helps to heuristically improve the Voronoi cell-based CVP algorithms [27]. Yet their heuristic correctness and analysis rely on the presence of the *shortest* vectors from $\Lambda_q^\perp(\mathbf{C})$, which, as we believe, the preimage ISIS queries do not help to obtain fast.

III. Dual counterpart to one-more-ISIS: one-more-LWE. As LWE can be seen as the 'lattice dual' of SIS, it is tempting to find a one-more-LWE definition that would be 'lattice dual' to one-more-ISIS, with hopefully bi-directional reductions between one-more-ISIS and one-more-LWE. This dual to one-more-ISIS could possibly shed light on the computation hardness of one-more-ISIS.

We propose the following one-more-LWE definition, and leave it as an open problem to study its relationship to one-more-ISIS. The attacker is given as input a matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ and arbitrarily many vectors $\mathbf{t}_i \in \mathbb{Z}_q^n$ of the form $\mathbf{t}_i = \mathbf{s}_i^t\mathbf{C} + \mathbf{e}_i^t$ with $\mathbf{e}_i$ short. The attacker is given access to an LWE oracle that on input $\mathbf{t}'_j$ (not necessarily among the input $\mathbf{t}_i$'s) returns $\mathbf{s}_j$ and $\mathbf{e}_j$ such that $\mathbf{s}_j^t\mathbf{C} + \mathbf{e}_j^t = \mathbf{t}'_j \bmod q$, if such a pair $(\mathbf{s}_j, \mathbf{e}_j)$ exists with a short $\mathbf{e}_j$. If ℓ is the number of LWE oracle queries, the attacker must output $\ell + 1$ pairs $(\mathbf{t}_i, \mathbf{s}_i)$ (with vectors $\mathbf{t}$ among the inputs).

Acknowledgments. We thank Olivier Blazy, Sébastien Canard, Carmit Hazay, Adeline Roux-Langlois and Muthuramakrishnan Venkitasubramaniam for insightful discussions. This work was partly supported by the DST "Swarnajayanti" fellowship, an IndoFrench CEFIPRA project, National Blockchain Project, the CCD Centre of Excellence, European Union Horizon 2020 Research and Innovation Program Grant 780701, BPI-France in the context of the national project RISQ (P141580), and the ANR AMIRAL project (ANR-21-ASTR-0016). Elena Kirshanova is supported by the Young Russian Mathematics scholarship and by the Russian Science Foundation grant N 22-41-04411, https://rscf.ru/project/22-41-04411/. Part of the research corresponding to this work was conducted while the first three authors were visiting the Simons Institute for the Theory of Computing.

References

1. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In *EUROCRYPT*, 2001.
2. M. Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996.
3. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, 1999.
4. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, 2001.
5. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In *Financial Crypto*, 2020.
6. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. On lattice-based interactive protocols: An approach with less or no aborts. In *ACISP*, 2020.
7. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In *USENIX Security*, pages 327–343, 2016.
8. A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe, and A. Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symmetric Cryptol.*, 2020.
9. S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *ACM SIGSAC*, 2017.
10. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem (shortened version). In *STACS*, 1985.
11. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, 2014.
12. A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, 2016.
13. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *J. Cryptol.*, 2003.
14. E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT*, 2019.
15. F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)security of ROS. In *EUROCRYPT*, 2021.
16. O. Blazy, P. Gaborit, J. Schrek, and N. Sendrier. A code-based blind signature. In *ISIT*, 2017.
17. A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In *PKC*, 2003.
18. D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *PKC*, 2011.
19. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *ASIACRYPT*, 2001.
20. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO*, 2019.
21. S. Bouaziz-Ermann, S. Canard, G. Eberhart, G. Kaim, A. Roux-Langlois, and J. Traoré. Lattice-based (partially) blind signature without restart. *IACR Cryptol. ePrint Arch.*, 2020.
22. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.
23. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.
24. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO*, 1992.
25. N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *ASIACRYPT*, 2001.
26. D. Derler, S. Ramacher, and D. Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto*, 2018.
27. E. Doulgerakis, T. Laarhoven, and B. de Weger. Finding closest lattice vectors using approximate Voronoi cells. In *PQCrypto*, 2019.
28. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018.
29. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT*, 2020.
30. M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, and D. Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS*, 2019.
31. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *CRYPTO*, 2006.

32. P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Technical report. Specification available at https://falcon-sign.info/.
33. G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed elgamal encryption in the algebraic group model. In *EUROCRYPT*, 2020.
34. S. Garg and D. Gupta. Efficient round optimal blind signatures. In *EUROCRYPT*, 2014.
35. S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signatures. In *CRYPTO*, 2011.
36. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
37. L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In *USENIX Security*, 2021.
38. T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, 2012.
39. G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, 2011.
40. E. Hauck, E. Kiltz, and J. Loss. A modular treatment of blind signatures from identification schemes. In *EUROCRYPT*, 2019.
41. E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-based blind signatures, revisited. In *CRYPTO*, 2020.
42. S. Ibrahim, M. Kamat, M. Salleh, and S. A. Aziz. Secure E-voting with blind signature. In *NCTT*, 2003.
43. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO*, 1997.
44. J. Kastner, J. Loss, and J. Xu. On pairing-free blind signature schemes in the algebraic group model. *IACR Cryptol. ePrint Arch.*, 2020.
45. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, 2000.
46. H. Q. Le, W. Susilo, T. X. Khuc, M. K. Bui, and D. H. Duong. A blind signature from module lattices. In *DSC*, 2019.
47. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, 2013.
48. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.
49. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Efficient lattice-based blind signatures via gaussian one-time signatures. In *PKC*, 2022.
50. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *PKC*, 2021.
51. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.
52. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 2007.
53. P. Q. Nguyên and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2008.
54. M. Ohkubo and M. Abe. Security of some three-move blind signature schemes reconsidered. In *SCIS*, 2003.
55. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, 1992.
56. D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. *IET Information Security*, 2019.
57. C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *CRYPTO*, 2008.
58. A. Petzoldt, A. Szepieniec, and M. S. E. Mohamed. A practical multivariate blind signature scheme. In *Financial Crypto*, 2017.
59. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptol.*, 2000.
60. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 2009.
61. M. Rückert. Lattice-based blind signatures. In *ASIACRYPT*, 2010.
62. C.-P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, 1994.
63. J. Stern. A new paradigm for public key identification. *IEEE Trans. Inf. Theory*, 1996.
64. S. Tessaro and C. Zhu. Short pairing-free blind signatures with exponential security. *IACR Cryptol. ePrint Arch.*, 2022.
65. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO*, 2019.
66. X. Yi and K.-Y. Lam. A new blind ECDSA scheme for bitcoin transaction anonymity. In *Asia-CCS*, 2019.
67. C. Yin, S. Huang, P. Su, and C. Gao. Secure routing for large-scale wireless sensor networks. In *ICCT*, 2003.