Can Round-Optimal Lattice-Based Blind Signatures be Practical?

Shweta Agrawal* Elena Kirshanova† Damien Stehlé‡ 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 make advances towards making lattice-based blind signatures practical. We introduce two round-optimal constructions in the random oracle model, and provide guidance towards their concrete realization as well as efficiency estimates.

The first scheme relies on the homomorphic evaluation of a lattice-based signature scheme. This requires an HE-compatible lattice-based signature. For this purpose, we show that the rejection step in Lyubashevsky's signature is unnecessary if the working modulus grows linearly in $\sqrt{Q}$, where Q is an a priori bound on the number of signature queries. Compared to the state of art scheme from Hauck *et al* [CRYPTO'20], this blind signature compares very favorably in all aspects except for signer cost. Compared to a lattice-based instantiation of Fischlin's generic construction, it is much less demanding on the user and verifier sides.

The second 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. The resulting blind signature beats all the aforementioned from most angles and obtains practical overall performance.

1 Introduction

Blind signatures are a fundamental cryptographic primitive with numerous applications in e-cash [23], e-voting [48] cryptocurrencies [77] and other privacy-preserving technologies [78]. 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 several interactions with the signer. While there exist many practical realizations of blind signatures under number-theoretic assumptions (see, e.g., [64, 65, 66, 67, 71, 61]), the situation is very unsatisfactory in the post-quantum regime, especially in the context of optimal round complexity. Below, we summarize the state of the art from lattice assumptions, which is the focus of our work.

State of the Art from Lattices. The first proposal of a lattice-based blind signature was by Rückert [70], which has been investigated in a series of works, all of which have been shown to have incomplete security proofs and sometimes even attacks [70, 6, 5, 20, 52, 62] – we refer to [46] for a detailed discussion. The state of the art construction in this regime is due to Hauck *et al* [46], who state that the main goal of their work "is to give the first direct lattice-based blind signature scheme with a correct security proof". However,

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

despite being in the Random Oracle Model (ROM), this construction suffers from some significant drawbacks impacting its efficiency:

1. *Noise Flooding.* The construction relies on a variant of noise flooding, used with the Short Integer Solution problem (SIS). For λ-bit security, it leads to setting the SIS parameters such that $n \log q = \widetilde{\Omega}(\lambda^3)$, which is a lower bound on the signature bit-length. More precisely, the noise flooding derives from the requirement that the lattice-based linear hash function must be (ε, Q)-regular (see [46, Section 3]) with an ε that is $2^{-\Omega(\lambda)}$ for [46, Theorem 1] to be applicable to adversaries succeeding with probability $2^{-o(\lambda)}$. This is achieved in [46, Section 6] by considering two balls of the same radius that is $2^{\Omega(\lambda)}$ larger than the offset between the ball centers.

2. *Loss in Security Proof.* The scheme is an adaptation of the Okamoto-Schnorr blind signature [61] from the discrete logarithm setting to the lattice setting. The best known security proof for the Okamoto-Schnorr blind signature under standard assumptions [67], suffers from a loss in advantage larger than $2^Q/|\mathcal{C}|$. Here Q denotes an upper bound on the number Q_S of generated signatures and $\mathcal{C}$ denotes the challenge space. Note that $\log |\mathcal{C}|$ is a lower bound on the signature bit-length. This advantage loss stems from the ability of a malicious user to perform concurrent protocol executions with the signer and was recently shown to be intrinsic [16]. In the lattice setting, the proof from [46] suffers from a similar loss, which drastically limits Q or forces to have signature bit-lengths that grow at least linearly in Q (note that no devastating attack as the one from [16] is known for the scheme [46], so a better proof could possibly exist).

3. *Suboptimal Round Complexity.* Finally, the scheme of [46] uses three rounds, while optimal round complexity is two.

Due to the above factors, by parameters provided in [46], the constructed blind signature has size $\approx$ 7.73MB, for security against adversaries limited to getting 7 signatures. Thus, while the work makes an important contribution by providing a correct lattice based blind signature, it may be seen only as a feasibility result. Since blind signatures are practice oriented primitives, there remains a large distance to be covered until an acceptable solution.

Taking a Step Back. In this work, we take a step back and ask whether lattice based blind signatures can be practical. Looking more closely at the Hauck *et al* construction, an immediate observation is that the advantage loss stems from the ability of a malicious user to perform concurrent protocol executions with the signer. This loss was recently shown to be intrinsic [16], at least in the discrete logarithm setting. In turn, the ability of the user to engage in concurrent protocol executions stems from the fact that the protocol has three rounds: this attack could not arise in a two round protocol. Needless to say, smaller round complexity is desirable even for its own sake, and in the context of blind signatures, round complexity of two is both achievable and optimal [37, 40].

There are two main solutions in the literature for round optimal blind signatures which can be instantiated from lattice assumptions: one by Fischlin [37] in the CRS model, and one by Garg *et al* in the standard model [40]. However, both have primarily been interpreted as feasibility results as they make use of heavyweight techniques. In more detail, the work of Fischlin uses general purpose non-interactive zero knowledge proofs (NIZK), while the work of Garg *et al* relies heavily on complexity leveraging and general purpose witness indistinguishable proofs. Indeed, the subsequent work of [39] was dedicated to mitigating the impact of complexity leveraging and instantiating the proofs to minimize costs, but this work is restricted to the number-theoretic setting.

1.1 Our Contributions

We reopen the possibility of constructing practical two round blind signatures from lattices. As a starting point, we adapt Fischlin's scheme [37] to the ROM and instantiate it with efficient lattice based signatures and NIZKs. Due to the extensive research in efficient lattice based signatures [41, 55, 36, 44, 38, 11, 31] and

proof systems [53, 29, 15, 76, 19, 36, 35, 56] over the last 15 years, this already provides a candidate which is "somewhat reasonable" in practice. Using this along with [46] as benchmarks, we then provide two new constructions:

1. *From Homomorphic Encryption:* We provide a two-round construction based on homomorphic encryption (HE) in the ROM, which builds upon the multi-party computation (MPC) based construction of [40]. Our construction enjoys the following advantages:

   - *Efficiency.* Compared to [46], we reduce the amount of noise flooding from $2^{\Omega(\lambda)}$ down to $\sqrt{Q}$, where Q is the bound on the number of generated signatures and λ is the security parameter.
   - *Number of Signing Queries.* Unlike the scheme from [46], our construction enjoys a proof that does not suffer from a loss that is exponential in Q. Using lattice hardness assumptions over rings, we obtain signatures of bit-lengths bounded as $\widetilde{O}(\lambda \log^2 Q)$. In contrast, the signature bit-length in the scheme from [46] is $\Omega(\lambda^3 + Q \cdot \lambda)$.
   - *Computational Cost.* In terms of verification cost, our scheme outperforms the one of [46]. On the other hand, the comparison is more complex for signing cost. Our signing algorithm requires HE evaluation of a hash function. However, we show that using suitable building blocks, our signing cost can be reduced to several minutes with appropriate parallelization (see Table 1). We note that though the signing algorithm from [46] is simpler, if we adjust its parameters to support as many signatures as we do, the bit-sizes may grow so much that even their basic operations may become noncompetitive compared to our more complex operations on small data.

     Compared to the lattice adaptation of Fischlin's scheme, our construction offers a different tradeoff. The signatures in our HE based construction are shorter, and user complexity is significantly better. However, signer complexity is higher for the HE based construction, and the signature size depends on an a-priori bound on the number of queries, unlike in the adaptation of Fischlin's scheme. We note that in practical applications, we expect the signer to be a server with much higher computational resources than the user, so it may often be desirable to favour the user.

   Our construction also has the following disadvantages:

   - *Very Honest Signer Model:* The main limitation of our HE based construction is that it achieves somewhat reasonable parameters only in the "very honest signer" model as against the more standard "honest signer" model, where the latter imposes that the signing keys are generated honestly while the former further assumes that the signer behaves honestly during protocol execution. In fact, as discussed in Subsection 4.2, the cost of upgrading the HE-based construction to honest signer blindness is likely to be prohibitive using off the shelf components. Mitigating this cost is an important direction for future work.
   - *Large Communication Complexity:* Another limitation of our HE based approach is the large communication complexity (about 50 MB). However, though this is large, we observe that all of this data except for a few kB is independent from the message μ to be signed. The corresponding computation and transmission can hence be off-lined and amortized across several protocol executions (including with different signers).

2. *From One-More-ISIS:* We introduce a new assumption, one-more-ISIS, from which we construct a two round blind signature with very efficient parameters. Informally, the one-more-ISIS assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell + 1$ GPV signatures [41], even when given access to up to ℓ inversions of arbitrary syndromes.

   This construction supports an unbounded number of signatures, does not use noise flooding, and achieves overall superior computational cost than all the aforementioned candidates (although other candidates may outperform it in individual parameters). On the other hand, unlike the other constructions, 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, and we provide detailed cryptanalysis to justify it.

At a high level, this new candidate is similar in spirit to Fischlin's [37] and Chaum's [23]. While Fischlin's construction uses general purpose zero knowledge proofs, we can instead rely on algebraic proofs for linear relations [35, 56], which can be much more efficient. This lets us reduce the signature size as well as the cost of generating and verifying the signature.

Table 1 provides a comparison between the three round optimal approaches. The contents of the table are only rough estimates, detailed in the corresponding sections. Precise numbers can only be provided with implementations, which is beyond the scope of this work.

Construction	**Sig Size**	**User Time**	**Signer Time**	**Transcript Size**	**Verifier Time**
Fischlin, Sec. 3	$\sim$ 130 kB	up to 1h	$<$ 1 ms	$\sim$ 5 kB	Few secs
HE, Sec. 4	3-8 kB	$<$ 1 ms (online) Few mins (offline)	$\sim$ 200 min (w.o. parallelization)	$\sim$ 200 kB (online) $\sim$ 50 MB (offline)	Few ms
one-more-ISIS, Sec. 5	30-100 kB	$<$ 100 ms	$<$ 1 ms	$<$ 2 kB	Few ms

Table 1 Comparison of two round blind signature schemes. Parameters for the HE based scheme are in the "very honest signer" model.

1.2 Technical Overview

Protocol Based on HE. We believe that given the recent advances in HE and NIZKs, this approach may be a viable design strategy towards practical constructions of blind signatures from lattices. When starting from [40], making this approach work requires several new ideas, which we describe next.

Adapting Protocol by Garg et al *[40].* We revisit the construction of [40] and observe that if it is *degraded* to rely on the ROM (its primary objective was to provide a standard model construction), then the primary source of inefficiency, namely complexity leveraging, can be removed. Similarly, expensive witness indistinguishable ZAPs can be replaced by simple and efficient zero-knowledge arguments for specific algebraic statements [54, 55] by using ROM. We instantiate two round MPC of their construction using HE.

A question is how to instantiate the HE scheme with the required properties: our protocol needs malicious circuit privacy (Definition 2.4) and algebraic structure which allows for efficient non-interactive proofs for proving well-formedness of the public and secret key. To avoid using general zero knowledge proofs, we instantiate the HE scheme using TFHE [25, 26]. With this choice, it suffices to prove statements of linear relations with short unknowns, for which specific lattice-based techniques are known (see [76, 19, 35, 56], and references therein). These techniques result in much shorter proof sizes as well as much more efficient running times for the prover and verifier. We can also use techniques of [34] to upgrade TFHE to satisfy semi-honest circuit privacy and then leverage the ROM to upgrade it to the malicious setting, in an efficient way.

Circuitizing the Underlying Signature. Our blind signature requires to evaluate a signing algorithm under the hood of an HE scheme. For this to be efficient, we need a lattice-based signature scheme that can be expressed as a relatively simple circuit. The GPV signature scheme [41] and its practical versions [33, 68, 38] seem ill-suited to our needs, as the signing algorithm is very sequential, and the required 1-dimensional Gaussian samples are obtained via algorithms based on rejection sampling (see, e.g., [47, 79]) that are costly to transform into circuits. The other candidate is Lyubashevsky's signature scheme [54, 55]. It has the advantage of being far less sequential, but it also relies on rejection sampling: when some rejection test does not pass, then one needs to restart the signing process. The rejection probability is typically a non-zero constant: for example, it is ≈ 0.85 for the recommended parameter set of [31].

We show that Lyubashevsky's signature is suitable for transformation into a circuit using two modifications which enjoy different performance properties. The first possibility is to run $Q \cdot \omega(\log \lambda)$ executions of Lyubashevsky's signing algorithm in parallel and keep one that passes the rejection test (return $\perp$ if they all fail). If no more than Q signatures are generated, then for each of them the signing algorithm succeeds with probability $\geq 1 - \lambda^{-\omega(1)}$. However, this approach requires parallel repetition, implementing the rejection test and choosing the right output which may be cumbersome to implement homomorphically, and additionally results in some error of correctness. Another possibility is to remove the rejection test from Lyubashevsky's signature altogether, which we describe next. Observe that Lyubashevsky's signature consists of a pair of matrices or polynomials over the integers (depending whether one relies on the SIS assumption or its ring variant). It is well-known that when the magnitudes of these integers is bounded by $\lambda^{O(1)}$, then removing the rejection test leads to polynomial-time attacks. Oppositely, taking exponential magnitudes allows, via the noise flooding technique, to prove unforgeability of the rejection-free variant [28], but this leads to setting $n \log q = \widetilde{\Omega}(\lambda^3)$. In this work, we show that limited magnitudes of the order of $\sqrt{Q}$ suffice to ensure unforgeability. The proof is based on the Rényi divergence technique [57, 51, 12] and results in a sufficient condition $n \log q = \widetilde{\Omega}(\lambda \log^2 Q)$. Since Q is often a moderate polynomial in practice, this improvement is significant. Furthermore, we show that this amount of noise flooding is optimal by exhibiting a statistical attack if smaller noise is used. We believe this contribution could have other applications, for instance in evaluating this signature within an MPC protocol.

Protocol Based on one-more-ISIS. The starting point of our protocol based on the one-more-ISIS assumption is Fischlin's protocol [37], which we instantiate in Section 3. Its primary source of inefficiency is the use of general purpose NIZKs. In our lattice adaptation using GPV "Hash and Sign" signatures [41], 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 [15, 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. For most applications, it may be assumed that the signer is more powerful than the user, and choosing the tradeoff in computation cost to favour the signer over the user does not seem meaningful.

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 $(\mathbf{C}, \mathbf{A}, \mathsf{PKE.pk})$, ct and μ, there exists r and vectors $\mathbf{x}, \mathbf{y}$ such that

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

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 [69] (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-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-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 [14]. 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 $\mathbf{A} \cdot \mathbf{e}_i = 0 \bmod q$, find a short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{A} \cdot \mathbf{e} = 0 \mod q$, 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* [17], relying on the CFS signature scheme [27] and Stern zero-knowledge proofs [74]. 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 use of Stern proofs, their construction achieves signature size of several MB. A blind signature scheme based on multivariate polynomial systems was described in [63], with a non-standard unforgeability security property.

2 Preliminaries

In this section, we provide several 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), signatures and commitments. 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 poly(λ) 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 Homomorphic Encryption (HE)

A homomorphic encryption scheme is an encryption scheme that allows computations on encrypted data.

Definition 2.1 (Homomorphic Encryption)**.** A homomorphic encryption scheme HE is a tuple of PPT algorithms $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Eval}, \mathsf{HE.Dec})$ defined as follows:

- $\mathsf{HE.KeyGen}(1^\lambda, 1^d) \to (\mathsf{pk}, \mathsf{sk})$: On input the security parameter λ and a depth bound d, the KeyGen algorithm outputs a key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{HE.Enc}(\mathsf{pk}, \mu) \to \mathsf{ct}$: On input a public key pk and a message $\mu \in \{0,1\}$, the encryption algorithm outputs a ciphertext ct.
- $\mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}, \mathsf{ct}_1, \ldots, \mathsf{ct}_k) \to \hat{\mathsf{ct}}$: On input a public key pk, a circuit $\mathcal{C} : \{0,1\}^k \to \{0,1\}$ of depth at most d, and a tuple of ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$, the evaluation algorithm outputs an evaluated ciphertext $\hat{\mathsf{ct}}$.
- $\mathsf{HE.Dec}(\mathsf{pk}, \mathsf{sk}, \hat{\mathsf{ct}}) \to \hat{\mu}$: On input a public key pk, a secret key sk and a ciphertext $\hat{\mathsf{ct}}$, the decryption algorithm outputs a message $\hat{\mu} \in \{0, 1, \perp\}$.

The definition above can be adapted to handle plaintexts over larger sets than $\{0,1\}$. Note that the evaluation algorithm takes as input a (deterministic) circuit rather than a possibly randomized algorithm. An HE should satisfy the compactness, correctness and security properties defined below.

Definition 2.2 (Compactness)**.** We say that an HE scheme is compact if there exists a polynomial function $f(\cdot, \cdot)$ such that for all λ, depth bound d, circuit $\mathcal{C} : \{0,1\}^k \to \{0,1\}$ of depth at most d, and $\mu_i \in \{0,1\}$ for $i \in [k]$, the following holds: for $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda, 1^d)$, $\mathsf{ct}_i \leftarrow \mathsf{HE.Enc}(\mathsf{pk}, \mu_i)$ for $i \in [k]$, $\hat{\mathsf{ct}} \leftarrow \mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$, the bit-length of $\hat{ct}$ is at most $f(\lambda, d)$.

Definition 2.3 (Correctness)**.** We say that an HE scheme is correct if for all λ, depth bound d, circuit $\mathcal{C} : \{0,1\}^k \to \{0,1\}$ of depth at most d, and $\mu_i \in \{0,1\}$ for $i \in [k]$, the following holds: for $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda, 1^d)$, $\mathsf{ct}_i \leftarrow \mathsf{HE.Enc}(\mathsf{pk}, \mu_i)$ for $i \in [k]$, $\hat{\mathsf{ct}} \leftarrow \mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$, we have

$$\Pr[\mathsf{HE.Dec}(\mathsf{pk}, \mathsf{sk}, \hat{\mathsf{ct}}) = \mathcal{C}(\mu_1, \ldots, \mu_k)] = 1 - \lambda^{-\omega(1)}.$$

Definition 2.4 (Security)**.** We say that an HE scheme is secure if for all λ and depth bound d, the following holds: for any adversary $\mathcal{A}$ with run-time $2^{o(\lambda)}$, the following experiment outputs 1 with probability $2^{-\Omega(\lambda)}$:

1. On input the security parameter λ and a depth bound d, the challenger runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda, 1^d)$ and $\mathsf{ct} \leftarrow \mathsf{HE.Enc}(\mathsf{pk}, b)$ for $b \leftarrow \{0,1\}$; it provides $(\mathsf{pk}, \mathsf{ct})$ to $\mathcal{A}$.
2. Adversary $\mathcal{A}$ outputs a guess b'; the challenger outputs 1 if $b = b'$.

In our application, we will also need the HE scheme to support circuit privacy, as defined below.

Definition 2.5 (Circuit Privacy)**.** We say that the homomorphic encryption scheme HE is semi-honest circuit private if for $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\lambda, 1^d)$, any circuit $\mathcal{C} : \{0,1\}^k \to \{0,1\}$ of depth at most d, $\mu_i \in \{0,1\}$ for $i \in [k]$, no $2^{o(\lambda)}$-time adversary can distinguish between the following distributions with $2^{-o(\lambda)}$ advantage:

$$\Big(\mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}, \{\mathsf{ct}_i\}_{i \le k}), \{\mathsf{ct}_i\}_{i \le k}, \mathsf{pk}, \mathsf{sk}\Big) \text{ and } \Big(\mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}^0, \{\mathsf{ct}'_i\}_{i \le k}), \{\mathsf{ct}_i\}_{i \le k}, \mathsf{pk}, \mathsf{sk}\Big),$$

where $\mathsf{ct}_i \leftarrow \mathsf{HE.Enc}(\mathsf{pk}, \mu_i)$ for $i \in [k]$, $\mathsf{ct}'_1 = \mathsf{HE.Enc}(\mathsf{pk}, \mathcal{C}(\mu_1, \ldots, \mu_k))$, $\mathsf{ct}'_i = \mathsf{HE.Enc}(\mathsf{pk}, 0)$ for $1 < i \le k$ and $\mathcal{C}^0 : \{0,1\}^k \to \{0,1\}$ is the trivial circuit of depth d that outputs its first input and ignores the others.

We say that HE is maliciously circuit private if the above holds even if the keys $(\mathsf{pk}, \mathsf{sk})$ and ciphertexts ct_i for $i \in [k]$ are not necessarily generated honestly. In this case, the μ_i's are defined as $\mu_i = \mathsf{HE.Dec}(\mathsf{sk}, \mathsf{ct}_i)$ for $i \in [k]$, where we assume without loss of generality that decryption always outputs a bit (even when the ciphertext is not well-formed).

2.2 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 2.6 (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 [49].

Definition 2.7 (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 2.8 (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. We also consider the notion of **very honest signer blindness** for which the signer $\mathcal{S}^*$ follows the protocol honestly.

2.3 Non-Interactive Zero Knowledge Arguments

Definition 2.9 (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 2.10 (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 $\mathsf{NIZKAoK}$.

2.4 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

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

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

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

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 $\mathbf{c}$.

2.5 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 [69, 22] when it is appropriately parameterized.

Definition 2.11 (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 2.12 (Short Integer Solution ($\mathsf{SIS}_{q,n,m,\beta}$))**.** 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 $\|\mathbf{v}\| \le \beta$ 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, 59, 41], when it is appropriately parameterized. The same holds for the *inhomogeneous* version of the SIS problem stated below.

Definition 2.13 (Inhomogeneous Short Integer Solution ($\mathsf{ISIS}_{q,n,m,\beta}$))**.** 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 $\|\mathbf{v}\| \le \beta$ and $\mathbf{A} \cdot \mathbf{v} = \mathbf{t} \bmod q$.

2.6 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, 41, 58], whereas the second is derived from [50, 41, 22].

Lemma 1. Let q, n, m be positive integers with $q \ge 2$ and $m \ge 6n\log_2 q$.

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 $\|\mathbf{e}\| \le \sigma\sqrt{m}$.

We also need the so-called "gadget" matrix, defined next.

Definition 2.14 (Gadget Matrix)**.** We say that a matrix $\mathbf{G} \in \mathbb{Z}_q^{k \times \ell}$ is a gadget matrix if there exists an efficient deterministic procedure $\mathbf{G}^{-1}$, which, on input $\mathbf{x} \in \mathbb{Z}_q^k$, output a matrix $\mathbf{G}^{-1}(\mathbf{x})$ with small norm such that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$. A common choice of the gadget matrix is the following "power-of-b" matrix, where the base b is a small integer (say $b = 2$). Let $\mathbf{G} = \mathbf{I}_k \otimes \mathbf{g}^\intercal \in \mathbb{Z}_q^{k \times k\lceil \log_b q \rceil}$ with $\mathbf{g}^\intercal = (1, b, \ldots, b^{\lceil \log_b q \rceil - 1})$ (implicitly setting $\ell = k\lceil \log_b q \rceil$). The $\mathbf{G}^{-1}$ function is then the base-b decomposition function. By default we will consider the "power-of-two" gadget matrix, but all our results apply with any matrix $\mathbf{G}$ with the following property: There exists a deterministic polynomial-time algorithm $\mathbf{G}^{-1}(\cdot)$, which on input $\mathbf{x} \in \mathbb{Z}_q^k$ outputs a "short" $\mathbf{c}$ such that $\mathbf{G}(\mathbf{c}) = \mathbf{x}$.

2.7 Rényi Divergence

The Rényi Divergence (RD) is a measure of closeness of any two probability distributions. In certain cases, especially in proving the security of cryptographic primitives where the adversary is required to solve a search-based problem, the RD can be used as an alternative to the statistical distance [12], which may help obtain security proofs for smaller scheme parameters and may sometimes lead to simpler proofs.

Definition 2.15 (Rényi Divergence)**.** Let P and Q be any two discrete probability distributions such that $Supp(P) \subseteq Supp(Q)$. Then for $a \in (1, \infty)$, the Rényi Divergence of order a is defined by

$$R_a(P||Q) = \left(\sum_{x \in Supp(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

For $a = 1$ and $a = \infty$, the RD is defined as

$$R_1(P||Q) = \exp\left(\sum_{x \in Supp(P)} P(x) \log \frac{P(x)}{Q(x)} \right) \text{ and } R_\infty(P||Q) = \max_{x \in Supp(P)} \frac{P(x)}{Q(x)}.$$

For any fixed distributions P and Q, the function $f(a) = R_a(P\|Q)$ is non-decreasing, continuous over $(1, \infty)$ and tends to $R_\infty(P\|Q)$ as a goes to infinity. Further, if $R_a(P\|Q)$ is finite for some a, then it tends to $R_1(P\|Q)$ as a tends to 1.

Lemma 2 ([12, Lemma 2.9])**.** Let $a \in [1, \infty]$. Let P and Q denote distributions with $Supp(P) \subseteq Supp(Q)$. Then the following properties hold

- **Log Positivity:** $R_a(P||Q) \geq R_a(P||P) = 1$.
- **Data Processing Inequality:** $R_a(P^f||Q^f) \leq R_a(P||Q)$ for any function f, where P^f (resp. Q^f) denotes the distribution of $f(y)$ induced by sampling $y \leftarrow P$ (resp. $y \leftarrow Q$).
- **Probability preservation:** Let $E \subseteq Supp(Q)$ be an arbitrary event. If $a \in (1, \infty)$, then

  $$Q(E) \geq P(E)^{\frac{a}{a-1}} / R_a(P\|Q).$$

  For $a = \infty$,

  $$Q(E) \geq P(E) / R_\infty(P\|Q).$$

  For $a = 1$, Pinsker's inequality gives the following analogue property:

  $$Q(E) \geq P(E) - \sqrt{\ln R_1(P||Q)/2}.$$

- **Multiplicativity:** Assume that P and Q are two distributions of a pair of random variables (Y_1, Y_2). For $i \in \{1, 2\}$, let P_i (resp. Q_i) denote the marginal distribution of Y_i under P (resp. Q), and let $P_{2|1}(\cdot|y_1)$ (resp. $Q_{2|1}(\cdot|y_1)$) denote the conditional distribution of Y_2 given that $Y_1 = y_1$. Then we have:

- $R_a(P||Q) = R_a(P_1||Q_1) \cdot R_a(P_2||Q_2)$ if Y_1 and Y_2 are independent for $a \in [1, \infty]$.
- $R_a(P||Q) \leq R_\infty(P_1||Q_1) \cdot \max_{y_1 \in Y_1} R_a(P_{2|1}(\cdot|y_1)||Q_{2|1}(\cdot|y_1))$.

- **Weak Triangle Inequality:** Let P_1, P_2, P_3 be three distributions with $Supp(P_1) \subseteq Supp(P_2) \subseteq Supp(P_3)$. Then we have

$$R_a(P_1||P_3) \leq \begin{cases} R_a(P_1||P_2) \cdot R_\infty(P_2||P_3), \\ R_\infty(P_1||P_2)^{\frac{a}{a-1}} \cdot R_a(P_2||P_3) \quad \text{if } a \in (1, +\infty). \end{cases} \tag{2.1}$$

We will use the following RD bounds. Note that proof tightness can often be improved by optimizing over a, as suggested in [75].

Lemma 3 ([51, Lemma 4.2])**.** For any n-dimensional lattice, $\Lambda \subseteq \mathbb{R}^n$ and $s > 0$, let P be the distribution $\mathcal{D}_{\Lambda,s,\mathbf{c}}$ and Q be the distribution $\mathcal{D}_{\Lambda,s,\mathbf{c}'}$ for some fixed $\mathbf{c}, \mathbf{c}' \in \mathbb{R}^n$. If $\mathbf{c}, \mathbf{c}' \in \Lambda$, let $\varepsilon = 0$. Otherwise fix $\varepsilon \in (0, 1)$ and assume that $s > \eta_\varepsilon(\Lambda)$. Then for any $a \in (1, +\infty)$

$$R_a(P||Q) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon} \right)^{\frac{2}{a-1}}, \left(\frac{1+\varepsilon}{1-\varepsilon} \right)^{\frac{2}{a-1}} \right] \cdot \exp\left(a\pi \frac{||\mathbf{c} - \mathbf{c}'||^2}{s^2} \right).$$

2.8 Other Useful Lemmas

Lemma 4 (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 $\{\mathbf{A} \in \mathbb{Z}_q^{n \times m} : \mathbf{r} \mapsto \mathbf{A}\mathbf{r}\}$ is 2-universal for any prime q.

The following lemma is adapted from [55], which uses a different Gaussian normalization. In our uses of the third item, for simplicity, we set $k = \sqrt{2/\pi}$, for which the probability upper bound is $\leq 2^{-m}$.

Lemma 5 (Adapted from [55, 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 [37]. This will serve as a point of comparison for the constructions we will present in the next sections.

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 1 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 5) 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 satisfies $\mathbf{Cy} = H(\mathsf{ct})$.
   - 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.

Figure 1 Adaptation of Fischlin's Blind Signature.

3.2 Security

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

Theorem 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

$H(c) = \mathbf{C}\mathbf{y}$. 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 4, 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.

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 5), 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 2. Assume that PKE is $\mathsf{IND\text{-}CPA}$ secure and the $\mathsf{NIZKAoK}$ is (computational) 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 encrypt 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 may not satisfy full-fledged blindness as stated. 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 PKE.pk is computationally indistinguishable from uniform, then we could replace 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 PKE.pk.

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 [38], which is an efficient instantiation of the TrapGen-SamplePre framework from [41]. 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 [38], 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 [15]. 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, 43].

4 Two Round Blind Signature from MPC

In this section, we study the possibility of building practical round-optimal lattice based signatures in the random oracle model, using HE.

4.1 HE-based Blind Signature

We describe our two-round construction of blind signatures in Figure 2, which is a significant simplification of the construction by [40]. Our construction uses the following building blocks:

- A UF-CMA signature scheme $\mathsf{Sig} = (\mathsf{Sig.KeyGen}, \mathsf{Sig.Sign}, \mathsf{Sig.Verify})$.
- A maliciously circuit private homomorphic encryption scheme (see Definition 2.4) $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ that is capable of homomorphically evaluating circuits with depth up to the depth of $\mathsf{Sig.Sign}_{\mathsf{Sig.sk},\rho}$, the circuit that takes μ as input and returns $\mathsf{Sig.Sign}(\mathsf{Sig.sk}, \mu; \rho)$ (i.e., with the key Sig.sk and the Sig.Sign randomness ρ being hardwired).

- A NIZKAoK for the well-formedness of an HE key-pair, i.e., that there exists (sk, r) such that $(\mathsf{pk}, \mathsf{sk}) = \mathsf{HE.KeyGen}(1^\lambda; r)$; here r denotes the randomness used by HE.KeyGen.

The construction is described in Figure 2.

Setup. $\mathsf{Gen}(1^\lambda)$: Upon input the security parameter λ, do the following:

- Invoke $(\mathsf{Sig.sk}, \mathsf{Sig.vk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$.
- Output $\mathsf{BSig.sk} = \mathsf{Sig.sk}$, $\mathsf{BSig.vk} = \mathsf{Sig.vk}$.

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

1. **User:** Given the key BSig.vk and a message μ, user $\mathcal{U}$ does the following:
   - It first samples HE.KeyGen randomness r and an HE key-pair $(\mathsf{pk}, \mathsf{sk}) = \mathsf{HE.KeyGen}(1^\lambda; r)$; it then generates a NIZKAoK π_{sk} that $(\mathsf{pk}, \mathsf{sk})$ is well-formed, i.e., that there exists a pair (sk, r) such that $(\mathsf{pk}, \mathsf{sk}) = \mathsf{HE.KeyGen}(1^\lambda; r)$.
   - It computes $\mathsf{ct}_\mu = \mathsf{HE.Enc}(\mathsf{pk}, \mu)$.
   - It sends $(\mathsf{pk}, \pi_{\mathsf{sk}}, \mathsf{ct}_\mu)$ to signer $\mathcal{S}$.
2. **Signer:** Upon receiving $(\mathsf{pk}, \pi_{\mathsf{sk}}, \mathsf{ct}_\mu)$, signer $\mathcal{S}$ does the following:
   - It verifies π_{sk} and outputs $\perp$ if it fails.
   - It samples Sig.Sign randomness ρ and computes a ciphertext $\mathsf{ct}_\sigma = \mathsf{HE.Eval}(\mathsf{pk}, \mathsf{Sig.Sign}_{\mathsf{Sig.sk}, \rho}, \mathsf{ct}_\mu)$ that decrypts to $\mathsf{Sig.Sign}(\mathsf{Sig.sk}, \mu; \rho)$.
   - It sends ct_σ to user $\mathcal{U}$.
3. **User:** Upon receiving ct_σ, user $\mathcal{U}$ computes and outputs $\sigma = \mathsf{HE.Dec}(\mathsf{sk}, \mathsf{ct}_\sigma)$.

Verifying. The verifier computes $\mathsf{Sig.Verify}(\mathsf{Sig.vk}, \mu, \sigma)$ and outputs the result.

Figure 2 HE-based Round-Optimal Blind Signature.

Completeness. The completeness of the blind signature is argued as follows. From the completeness of NIZKAoK, the signer accepts π_{sk} (and does not abort) if the user computed the HE public key and π_{sk} correctly. From the correctness of HE, the element $\mathsf{ct}_\sigma = \mathsf{HE.Eval}(\mathsf{Sig.Sign}_{\mathsf{BSig.sk}}, \mathsf{ct}_\mu)$ decrypts to $\mathsf{Sig.Sign}_{\mathsf{Sig.sk}}(\mu) = \mathsf{Sig.Sign}(\mathsf{Sig.sk}, \mu) = \sigma$. Hence $\mathsf{HE.Dec}(\mathsf{sk}, \mathsf{ct}_\sigma)$ outputs a signature $\sigma = \mathsf{Sig.Sign}(\mathsf{Sig.sk}, \mu)$. It passes verification, by correctness of Sig.

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

Theorem 3. Assume that the underlying signature scheme satisfies UF-CMA security, the homomorphic encryption scheme HE satisfies malicious circuit privacy and the NIZKAoK is knowledge sound. Then the construction in Figure 2 satisfies one more unforgeability.

Proof. The argument proceeds via the following hybrids.

Hybrid_0: This is the genuine one-more unforgeability experiment.

Hybrid_1: This hybrid differs from the previous one in that the challenger extracts sk from π_{sk}, and aborts if it fails to do so.

Hybrid_2: In this hybrid, the challenger uses sk to compute $\mu_i = \mathsf{HE.Dec}(\mathsf{sk}, \mathsf{ct}_i)$ for all $i \in [Q_S]$. It then changes the way the Q_S ciphertexts $\mathsf{ct}_{\sigma_1} \ldots, \mathsf{ct}_{\sigma_{Q_S}}$ are generated. It first signs the messages $\mu_1, \ldots, \mu_{Q_S}$ to obtain signatures $\sigma_1, \ldots, \sigma_{Q_S}$ (in the clear). Then, it encrypts $\sigma_1, \ldots, \sigma_{Q_S}$ using HE to obtain

$\mathsf{ct}'_{\sigma_1}, \ldots, \mathsf{ct}'_{\sigma_{Q_S}}$. It runs $\mathsf{HE.Eval}(\mathsf{pk}, \mathcal{C}^0, (\mathsf{ct}'_{\sigma_i}, \star))$ for all $i \in [Q_S]$ to obtain $\mathsf{ct}_{\sigma_1} \ldots, \mathsf{ct}_{\sigma_{Q_S}}$. Here $\mathcal{C}^0$ is a dummy circuit that has the same depth and number of inputs as the $\mathsf{Sig.Sign}$ signing circuit and outputs its first input and $\star$ represents as many independent encryptions of 0 as required (see Definition 2.5).

The indistinguishability between Hybrid_0 and Hybrid_1 follows from the knowledge soundness of the NIZKAoK. The indistinguishability between Hybrid_1 and Hybrid_2 follows from malicious circuit privacy of HE. Note here that semi-honest circuit privacy may not suffice, as we are not certain that the ciphertexts are properly generated (or even that sk is properly generated, as the NIZKAoK only guarantees its well-formedness). The theorem statement follows from Lemma 6, which shows that if Sig satisfies UF-CMA, then the adversary wins in Hybrid_2 only with negligible probability. □

Lemma 6. Assume that Sig satisfies UF-CMA security. Then the advantage of the adversary in the one more unforgeability game is negligible in Hybrid_2.

Proof. Assume there exists an adversary $\mathcal{U}^*$ that wins the one more unforgeability game in Hybrid_2. Then we build an adversary $\mathcal{B}$ against the UF-CMA security of the underlying signature scheme Sig. Adversary $\mathcal{B}$ does the following:

1. It obtains $\mathsf{Sig.vk}$ from the signature challenger, defines $\mathsf{BSig.vk} = \mathsf{Sig.vk}$ and forwards it to $\mathcal{U}^*$.
2. It runs the signing protocol with adversary $\mathcal{U}^*$ simulating the BS signer as follows:
   (a) Adversary $\mathcal{U}^*$ outputs the HE public key pk along with argument of knowledge π_{sk} of a corresponding sk. Adversary $\mathcal{B}$ rewinds $\mathcal{U}^*$ to extract sk from the NIZKAoK π_{sk}.
   (b) Adversary $\mathcal{U}^*$ also outputs Q_S ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_{Q_S}$. Adversary $\mathcal{B}$ uses sk to decrypt $\mathsf{ct}_1, \ldots, \mathsf{ct}_{Q_S}$ to obtain $\mu_1, \ldots, \mu_{Q_S}$.
   (c) Adversary $\mathcal{B}$ sends $\mu_1, \ldots, \mu_{Q_S}$ to the UF-CMA signature challenger and obtains signatures $\sigma_1, \ldots, \sigma_{Q_S}$.
   (d) It constructs $\mathsf{ct}_{\sigma_1} \ldots, \mathsf{ct}_{\sigma_{Q_S}}$ from $\sigma_1, \ldots, \sigma_{Q_S}$ as in the previous hybrid and returns these to $\mathcal{U}^*$.
   (e) When $\mathcal{U}^*$ outputs $Q_S + 1$ message-signature pairs (μ_i, σ_i) for $i \in [Q_S + 1]$ that pass verification and such that the μ_i's are distinct, it outputs any of these for which μ_i had not been queried to the UF-CMA signature challenger.

The success of $\mathcal{U}^*$ translates to the success of $\mathcal{B}$ in the UF-CMA game. □

We now argue that the HE-based blind signature BS satisfies very honest signer blindness.

Theorem 4. Assume that the argument systems are zero knowledge and that HE is semantically secure. Then the construction in Figure 2 satisfies very honest signer blindness.

Proof. The argument proceeds via a sequence of hybrids.

Hybrid_0: This is the genuine very honest signer blindness experiment.

1. The challenger generates $(\mathsf{BSig.sk}, \mathsf{BSig.vk})$ and returns these to the adversary $\mathcal{S}^*$.
2. The signer $\mathcal{S}^*$ outputs two messages μ_0 and μ_1. The challenger picks a uniform bit b.
3. The signer $\mathcal{S}^*$ interacts concurrently with $\mathcal{U}_0(\mathsf{BSig.vk}, \mu_b)$ and $\mathcal{U}_1(\mathsf{BSig.vk}, \mu_{\bar{b}})$, played by the challenger as follows:
   (a) User $\mathcal{U}_0$ (resp. $\mathcal{U}_1$) generates the HE public and secret keys $(\mathsf{pk}_0, \mathsf{sk}_0)$ (resp. $(\mathsf{pk}_1, \mathsf{sk}_1)$) honestly, along with arguments π_{sk_0} and π_{sk_1} of the well-formedness of the public keys.
   (b) Additionally, users $\mathcal{U}_0$ and $\mathcal{U}_1$ provide their respective ciphertexts $\mathsf{ct}_0 = \mathsf{HE.Enc}(\mathsf{pk}_0, \mu_b)$ and $\mathsf{ct}_1 = \mathsf{HE.Enc}(\mathsf{pk}_1, \mu_{\bar{b}})$.

(c) The challenger evaluates the signing algorithm homomorphically on ciphertexts ct_b and $\mathsf{ct}_{\bar{b}}$ to obtain ct'_0 and ct'_1, respectively (recall that this step is performed by the adversary in the honest signer blindness experiment, but by the challenger in the very honest signer blindness experiment); the challenger gives ct_b and $\mathsf{ct}_{\bar{b}}$ to the signer $\mathcal{S}^*$, as well as all intermediate values of their computation.

(d) After obtaining the evaluated HE ciphertexts, users $\mathcal{U}_0$ and $\mathcal{U}_1$ decrypt them using the HE secret keys sk_0 and sk_1 to obtain signatures σ_0 and σ_1 respectively.

(e) The signer $\mathcal{S}^*$ is given σ_0, σ_1.

(f) The signer $\mathcal{S}^*$ outputs its guess for bit b.

Hybrid_1: In this hybrid, the proofs π_{sk_0} and π_{sk_1} are replaced with simulated proofs.

Hybrid_2: In this hybrid, at Step 3(d), the signatures σ_0 and σ_1 are generated by using Sig.Sign directly on μ_b and $\mu_{\bar{b}}$, respectively.

Hybrid_3: In this hybrid, we replace $\mathsf{HE.Enc}(\mathsf{pk}_0, \mu_b)$ by $\mathsf{HE.Enc}(\mathsf{pk}_0, 0)$ and $\mathsf{HE.Enc}(\mathsf{pk}_1, \mu_{\bar{b}})$ by $\mathsf{HE.Enc}(\mathsf{pk}_1, 0)$.

Indistinguishability of hybrids

1. The indistinguishability between Hybrid_0 and Hybrid_1 follows from the zero-knowledge property of the underlying NIZKAoK.

2. The only difference between Hybrid_1 and Hybrid_2 is that in the former, we have $\sigma_0 = \mathsf{HE.Dec}(\mathsf{sk}_0, \mathsf{ct}'_0)$ (resp. $\sigma_1 = \mathsf{HE.Dec}(\mathsf{sk}_1, \mathsf{ct}'_1)$) while in the latter, the signatures σ_0 and σ_1 are generated using Sig.Sign directly. Indistinguishability follows from the correctness of HE and Sig.

3. The indistinguishability between Hybrid_2 and Hybrid_3 follows from the semantic security of HE. The proof is a standard reduction to the semantic security. Note that due to the previous hybrids, the blindness challenger does not need the HE secret key for any operations and can obtain the HE public keys and ciphertexts from the HE challenger. The arguments π_{sk} are simulated, and the signatures σ_b and $\sigma_{\bar{b}}$ are computed directly using the signing key Sig.sk.

To complete the proof of the theorem, it suffices to note that the advantage of the adversary in Hybrid_3 is 0 since the bit b is information theoretically hidden. □

In the next subsection, we explain that the scheme may only achieve very honest signer blindness, and discuss how to modify it to achieve the stronger notions of honest signer blindness as well as full-fledged blindness.

4.2 Upgrading Blindness

Here, we discuss how to upgrade the construction in Figure 2 to achieve the more standard notion of honest signer blindness. Recall that for honest signer blindness, the signer is not expected to adhere to the protocol during signing. Note that the scheme of Figure 2 may not satisfy the stronger notion of honest signer blindness, which differs from very honest signer blindness in that the signer can deviate from the protocol during the signing phase. For example, assume the signature scheme Sig is obtained by derandomizing a probabilistic signature scheme Sig_0, as follows: at key generation, one samples a PRF key k and appends it to the signing key; when signing a message, one first creates a pseudo-random string using k and the message to be signed, and then runs Sig_0.Sign with this pseudo-random string. Then, to break honest signer blindness, the signer could use two distinct PRF keys k_0 and k_1 for the two executions of the signing protocol. At the end, it could compute the four possible signatures from (μ_0, μ_1) and (k_0, k_1) and assess which PRF key has been used in which execution, by matching σ_0 and σ_1 with those four signatures.

To handle this, we first make the signature deterministic, to constrain what the signer can do. For this purpose, we use a PRF: during the setup phase, the signer samples a PRF key k that is now part of the blind

signature key $\mathsf{BSig.sk}$. During the signing phase, the signer generates a pseudo-random string ρ for $\mathsf{Sig.Sign}$ by using the PRF with key k, and input the ciphertext ct_μ. Taking ct_μ rather than μ as input to the PRF allows to avoid a possibly costly HE evaluation of the PRF.

To further constrain the signer's behavior, we additionally add a $\mathsf{NIZKAoK}$ π_σ on the signer's side, which shows that ct_σ is honestly computed using $\mathsf{BSig.sk}$, ct_μ and pk and that the same signing key $\mathsf{BSig.sk}$ is used across all signatures. To ensure this, we include a perfectly binding commitment of $\mathsf{BSig.sk}$ as part of $\mathsf{BSig.vk}$ and the signer generates a $\mathsf{NIZKAoK}$ for the following statement: Given $\mathsf{ct}_\sigma, \mathsf{ct}_\mu, \mathsf{pk}, \mathsf{com}$, there exist $\mathsf{BSig.sk}$ and r such that (i) $\mathsf{com} = \mathsf{Commit}(\mathsf{BSig.sk}, r)$, and (ii) $\mathsf{ct}_\sigma = \mathsf{HE.Eval}(\mathsf{pk}, \mathsf{Sig.Sign}_{\mathsf{Sig.sk}}, \mathsf{ct}_\mu)$. Upon receiving π_σ and ct_σ, the user first verifies π_σ, outputs $\perp$ if the verification fails and else continues as in the scheme of Figure 2. Unfortunately, from a concrete perspective, we do not see a way to implement this $\mathsf{NIZKAoK}$ at a moderate cost.

To upgrade from honest signer blindness to full-fledged blindness, the only further ingredient that we add is the requirement that the commitment is extractable. We can ensure this by instantiating the commitment with a perfectly correct PKE scheme. In the blindness proof, using the secret key of this PKE, the challenger can extract $\mathsf{BSig.sk}$ from the commitment.This allows to reduce to the honest signer blindness game above.

4.3 Making Lyubashevsky's Signature HE-friendly

In this section, we "flatten" the signing circuit of Lyubashevsky's signature scheme from [55] to make it more HE-friendly. As discussed in Section 1, we provide a rejection-free variant of Lyubashevsky's signature scheme from [55]. Our construction uses a hash function $H : \{0,1\}^* \to \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k; \|\mathbf{v}\|_1 \leq \alpha\}$, modeled as a random oracle. Here α is a parameter, typically much smaller than k.

The signature scheme is described in Figure 3.

$\mathsf{KeyGen}(1^\lambda)$: Upon input the security parameter λ, set $q, n, m, \beta, k, d, \sigma$ such $n = \Omega(\lambda)$ and the scheme is secure (see Theorem 5); then do the following:

1. Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{S} \leftarrow \{-d, \ldots, 0, \ldots, d\}^{m \times k}$.
2. Set $\mathbf{T} = \mathbf{AS}$.
3. Output $\mathsf{vk} = (\mathbf{A}, \mathbf{T})$, $\mathsf{sk} = \mathbf{S}$.

$\mathsf{Sign}(\mathsf{sk}, \mu)$: Upon input the signing key sk and a message μ, do the following:

1. Sample $\mathbf{y} \leftarrow \mathcal{D}_{\mathbf{Z}^m, \sigma}$.
2. Set $\mathbf{c} = H(\mathbf{Ay}, \mu)$.
3. Set $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$.
4. Output $(\mathbf{z}, \mathbf{c})$.

$\mathsf{Verify}(\mathsf{vk}, \mu, (\mathbf{z}, \mathbf{c}))$: Upon input the verification key vk, a message μ, and a signature $(\mathbf{z}, \mathbf{c})$, do the following:

1. Check if $\|\mathbf{z}\| \leq \gamma$, where $\gamma = (2\sigma + \alpha d)\sqrt{m}$.
2. Check if $H(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$.
3. If both checks pass, then accept, else reject.

Figure 3 Lyubashevsky's Signature Without Aborts

Correctness. Since $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$, where $\mathbf{y} \leftarrow \mathcal{D}_{\mathbf{Z}^m, \sigma}$, we have $\|\mathbf{z}\| \leq 2\sigma\sqrt{m} + \|\mathbf{Sc}\|$ with probability $1 - 2^{-\Omega(\lambda)}$, using standard Gaussian tail bounds (see, e.g., Lemma 5). Since $\|\mathbf{S}\|_\infty \leq d$ and $\|\mathbf{c}\|_1 \leq \alpha$, we have

$\|\mathbf{Sc}\| \leq d\alpha\sqrt{m}$. This gives us $\|\mathbf{z}\| \leq (2\sigma + d\alpha)\sqrt{m}$ with overwhelming probability. Finally, note that

$$H(\mathbf{Az} - \mathbf{Tc}, \mu) = H(\mathbf{A}(\mathbf{y} + \mathbf{Sc}) - \mathbf{ASc}, \mu) = H(\mathbf{Ay}, \mu) = \mathbf{c}.$$

Security. Next, we establish security via the following theorem.

Theorem 5. Assume that $m > \lambda + (n \log q)/\log(2d+1)$, $\sigma \geq \alpha d\sqrt{mQ}$ where Q is the maximum number of signing queries an attacker can make and $|D_H| \geq 2^\lambda$ where D_H is the range of H. Assume further that $\mathsf{SIS}_{q,n,m,\beta}$ is hard for $\beta = 2\gamma + 2d\alpha\sqrt{m}$. Then the construction in Figure 3 satisfies UF-CMA in the random oracle model.

Proof. We prove the security via the following hybrids:

Hybrid_0: This is the genuine security game, i.e., with honest executions of the Sign algorithm on signing queries by the adversary.

Hybrid_1: In this hybrid the challenger responds to the signing query for any message μ as follows.

1. Sample $\mathbf{y} \leftarrow \mathcal{D}_{\mathbf{Z}^m,\sigma}$ as in the previous hybrid.
2. Sample $\mathbf{c} \leftarrow \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \alpha\}$.
3. Set $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$.
4. Set $H(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$.
5. Output $(\mathbf{z}, \mathbf{c})$.

Hybrid_2: In this hybrid the challenger responds to the signing query for any message μ as follows.

1. Sample $\mathbf{c} \leftarrow \{\mathbf{v} : \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \alpha\}$.
2. Sample $\mathbf{z} \leftarrow \mathcal{D}_{\mathbf{Z}^m,\sigma}$.
3. Set $H(\mathbf{Az} - \mathbf{Tc}, \mu) = \mathbf{c}$.
4. Output $(\mathbf{z}, \mathbf{c})$.

The only difference between Hybrid_0 and Hybrid_1 is that in Hybrid_1, the output value for H is chosen at random, and then programmed as the answer to $H(\mathbf{Ay}, \mu)$ when a query for μ. By the definition of the random oracle, the two hybrids are indistinguishable.

The result now follows from the two claims below.

Claim 6. If there is an adversary that makes at most Q_S signing queries and can win the game in Hybrid_1 with probability δ, then its probability of winning in Hybrid_2 is polynomial in δ, if $\sigma \geq \alpha d\sqrt{mQ_S}$.

Proof. The only difference between the two hybrids is in the value of $\mathbf{z}$. For $i \in [Q_S]$, in Hybrid_1, we have $\mathbf{z}_i = \mathbf{y}_i + \mathbf{Sc}_i$ with $\mathbf{y}_i \leftarrow \mathcal{D}_{\mathbf{Z}^m,\sigma}$, while in Hybrid_2, we have $\mathbf{z}_i \leftarrow \mathcal{D}_{\mathbf{Z}^m,\sigma}$. Let us refer to the joint distribution of all $\mathbf{z}$'s in Hybrid_1 as D_1 and that in Hybrid_2 as D_2. Let E denote the event that the adversary wins the game. Then by our assumption, we have $D_1(E) = \delta$. From the probability preservation property (Lemma 2) of the Rényi Divergence, we get:

$$D_2(E) \geq \frac{\delta^{\frac{a}{a-1}}}{R_a(D_1 \| D_2)}, \text{ for any } a \in (1, \infty). \tag{4.1}$$

Computing $R_a(D_1\|D_2)$: For $i \in [Q_S]$, the vector $\mathbf{z}_i$ is from distribution $D_{1i} := \mathcal{D}_{\mathbf{Z}^m,\sigma,\mathbf{Sc}_i}$ in Hybrid_1 and from distribution $D_{2i} = \mathcal{D}_{\mathbf{Z}^m,\sigma}$ in Hybrid_2. Note that $D_1 = (D_{11}, \ldots, D_{1Q_S})$ and $D_2 = (D_{21}, \ldots, D_{2Q_S})$. By Lemma 3, we have

$$R_a[D_{1i}\|D_{2i}] = \exp\left(a\pi\frac{\|\mathbf{Sc}_i\|^2}{\sigma^2}\right) \quad \text{for any } a \in (1, \infty).$$

Recall from the correctness proof that we have $\|\mathbf{S}\mathbf{c}_i\| \leq d\alpha\sqrt{m}$. By using the multiplicativity property of the Rényi divergence (Lemma 2), we get:

$$R_a(D_1\|D_2) \leq \exp\left(a\pi\frac{Q_S(d\alpha\sqrt{m})^2}{\sigma^2}\right), \quad \text{for any } a \in (1,\infty). \tag{4.2}$$

Using the assumption $\sigma \geq d\alpha\sqrt{m}\cdot\sqrt{Q_S}$, we get $R_a(D_1\|D_2) \leq \exp(a\pi)$. Using Equation (4.1), we obtain that $D_2(E) \geq \delta^{\frac{a}{a-1}}\exp(-a\pi)$. Taking any value of $a > 1$ provides the result. □

Claim 7. If there is a forger $\mathcal{F}$ that makes at most Q_S signing queries and Q_H random oracle queries, and succeeds in forging a valid signature with probability δ in Hybrid_2, then we can define an algorithm $\mathcal{B}$ which given $\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}$, finds a non-zero $\mathbf{v}$ such that $\|\mathbf{v}\| \leq (2\gamma + 2d\alpha\sqrt{m})$ and $\mathbf{A}\mathbf{v} = \mathbf{0}$, with probability at least

$$\left(\frac{1}{2} - \frac{\varepsilon}{2}\right)\left(\delta - \frac{1}{|D_H|}\right)\left(\frac{\delta - 1/|D_H|}{Q_H + Q_S} - \frac{1}{|D_H|}\right).$$

This claim and its proof are identical to [55, Lemma 5.4]. Note that under the conditions of Theorem 5, the latter probability lower bound is $\geq \delta^2/(2(Q_H+Q_S)) - 2^{-\Omega(\lambda)}$. □

Note that the condition $\sigma \geq \alpha d\sqrt{mQ}$ from Theorem 5 forces to set a modulus q and a SIS bound β that grow linearly with $\sqrt{Q}$. To ensure λ bits of security, one may choose n growing linearly with $\sqrt{Q}$. Overall, if using a Ring-SIS or Module-SIS instantiation, then the bit-length of the signature grows linearly with $n\log q$ and hence with $\log^2 Q$.

Next, we show that the flooding noise used in the above construction is essentially optimal by exhibiting an attack when the flooding noise is smaller. In Section 4.3.2, we show how to handle uniform rather than Gaussian noise.

4.3.1 Optimality of Flooding

In this section, we show that the flooding amount used in the above construction is essentially optimal, and in particular that the dependence on $\sqrt{Q}$ is necessary. In more detail, we show that if the flooding noise is smaller than this, then an adversary can recover the signing key. Note that this attack is folklore, we recall it for the sake of completeness.

Statistical Attack. Recall that the signature for message M_i is of the form $(\mathbf{z}_i, \mathbf{c}_i)$, where $\mathbf{z}_i = \mathbf{S}\mathbf{c}_i + \mathbf{y}_i$, $\mathbf{c}_i \in \{-1,0,1\}^k$, $\|\mathbf{c}_i\|_1 \leq \alpha$, and $\mathbf{S}$ is the signing key. The adversary can obtain many such pairs corresponding to different messages. Let Q be the maximum number of signing queries that the adversary can make. Let $\mathbf{S}_i$ represents the ith row of matrix $\mathbf{S}$. Let $\mathbf{c}_{ij}$, $\mathbf{y}_{ij}$ and $\mathbf{S}_{ij}$ represent the jth entry in vectors $\mathbf{c}_i$, $\mathbf{y}_i$ and $\mathbf{S}_i$ respectively. Consider such tuples $(\mathbf{z}_i, \mathbf{c}_i)$ where $\mathbf{c}_{i1} = 1$. Let $B \subseteq [Q]$ be the set of such indices. The adversary gets approximately $Q/3$ such tuples corresponding to $i \in B$. For each i, using the first row of $\mathbf{S}$, we may write:

$$\mathbf{S}_{11} + \sum_{j=2}^{k}\mathbf{S}_{1j}\mathbf{c}_{ij} + \mathbf{y}_{i1} = \mathbf{z}_{i1} \tag{4.3}$$

We denote the average of $\sum_{j=2}^{k}\mathbf{S}_{1j}\mathbf{c}_{ij} + \mathbf{y}_{i1}$ over $i \in B$ as avg. We show that unless $\mathbf{y}_{i1}$ is $O(\sqrt{Q})$, we can recover $\mathbf{S}_{11}$. To conduct the attack, we bound each summand of avg separately.

Claim 8. Let $t_1 < 1/2$ be a positive constant and Q, k, d, α be as above. Then,

$$\Pr\left[\left|\frac{\sum_{i\in B}\sum_{j=2}^{k}\mathbf{S}_{1j}\mathbf{c}_{ij}}{|B|}\right| < t_1\right] \geq 1 - 2\exp\left(\frac{-Qt_1^2}{6(\alpha-1)^2d^2}\right)$$

Proof. Note that $\sum_{j=2}^{k} \mathbf{S}_{1j}\mathbf{c}_{ij}$ takes values in the range $[-(\alpha-1)d, (\alpha-1)d]$, with mean at 0. In more detail, let X be a random variable, with mean 0 and bound $[-(\alpha-1)d, (\alpha-1)d]$, then for some positive constant $t_1 < 1/2$, we have from Hoeffding's bound

$$\begin{aligned} & \Pr[|\bar{X} - E[X]| \geq t_1] && \leq 2\exp(\frac{-(Q/3)t_1^2}{2(\alpha-1)^2d^2}) \\ \implies & \Pr[|\bar{X}| \geq t_1] && \leq 2\exp(\frac{-Qt_1^2}{6(\alpha-1)^2d^2}) \\ \implies & \Pr[|\bar{X}| < t_1] && \geq 1 - 2\exp(\frac{-Qt_1^2}{6(\alpha-1)^2d^2}) \end{aligned}$$

Since d is small, in particular if $(6(\alpha-1)^2d^2 < Qt_1^2)$, then $1 - 2\exp(\frac{-Qt_1^2}{6(\alpha-1)^2d^2})$ is non-negligible. □

Let us assume that the average of $\mathbf{y}_{i1}$ is also smaller than $1/2 - t_1$ with non negligible probability. Then, $\mathsf{avg} < 1/2$ with non negligible probability. Summing both sides of Equation 4.3 over the set B, we get

$$\mathbf{S}_{11} + \mathsf{avg} = \frac{\sum_{i \in B} \mathbf{z}_{i1}}{|B|}$$

In this case the adversary can successfully recover $\mathbf{S}_{11}$ as

$$\mathbf{S}_{11} = \left\lfloor \frac{\sum_{i \in B} \mathbf{z}_{i1}}{|B|} \right\rceil$$

We now examine how large $\mathbf{y}_{i1}$ must be to avoid this attack. Let $Y \leftarrow \mathcal{D}_\sigma$ be the random variable representing the distribution of $\mathbf{y}_{i1}$ values. Then from Hoeffding's bound, for some constant c' and $t_2 < (1/2 - t_1)$,

$$\begin{aligned} & \Pr[|\bar{Y} - E[Y]| && \geq t_2] \leq 2\exp(-c'Qt_2^2/3\sigma^2) \\ \implies & \Pr[|\bar{Y}| && \geq t_2] \leq 2\exp(-c'Qt_2^2/3\sigma^2) \\ \implies & \Pr[\,|\bar{Y}| && < t_2] \geq 1 - 2\exp(-c'Qt_2^2/3\sigma^2) \end{aligned}$$

Thus, if $3\sigma^2 < c'Qt_2^2$, then $1 - 2\exp(-c'Qt_2^2/3\sigma^2)$ is non-negligible. Hence, for the average of $\mathbf{y}_{i1}$ to be greater than t_2, we need that $3\sigma^2 \geq c'Qt_2^2$, i.e. σ must grow proportional to $\sqrt{Q}$.

4.3.2 From Gaussian to Uniform Noise

In some applications, it may be preferable to use a vector $\mathbf{y}$ whose coordinates are uniform in some interval $[-B, B]$ rather than Gaussian (at Step 1 of the `Sign` algorithm). This is the choice made for the regular Dilithium signature scheme [31].

To adapt the current proof, the only step that needs to be modified is in the transition between Hybrid_1 and Hybrid_2. A difficulty is that the support of the distribution of $\mathbf{z}$ in Hybrid_2 has to contain the support of the distribution of $\mathbf{z}$ in Hybrid_1, for the Rényi divergence to be defined. For this purpose, we consider a wider interval in Hybrid_2, which contains all possible intervals $[-B, B]^m + \mathbf{Sc}$ of Hybrid_1. Concretely, in Hybrid_1, the vector $\mathbf{z}$ is sampled from $D_1 = U([-B, B]^m) + \mathbf{Sc}$, whereas in Hybrid_2, the vector $\mathbf{z}$ is sampled from $D_2 = U([-B', B']^m)$. As $\|\mathbf{Sc}\|_\infty \leq \alpha d$, we can take $B' = B + \alpha d$. Assuming that both B and B' are integers, we have

$$R_a(D_1 \| D_2) = \left(\frac{2B'+1}{2B+1}\right)^{Q_S m} \leq \left(1 + \frac{2\alpha d}{B}\right)^{Q_S m}, \quad \text{for any } a \in (1, \infty).$$

We obtain that for transiting from Hybrid_1 to Hybrid_2, it suffices to set $B \geq \Omega(md\alpha Q_S)$.

4.3.3 Using a Non-Cryptographic Hash.

Since the homomorphic evaluation of the hash function could be inefficient, it is natural to ask how complex the hash function in Figure 3 needs to be. In a recent result, Chen at al. [24] showed that Lyubashevsky's signature, which relies a cryptographic hash function H that can be modeled as a random oracle (such as SHA3-256), can be replaced by the bit decomposition function $\mathbf{G}^{-1}$ (defined in Section 2). This variant, which we discuss in Subsection 4.6, offers a different performance when computed homomorphically.

4.4 Efficiency Estimate

In this section, we analyze the efficiency of the blind signature from Section 4.1 if instantiated using the signature scheme in Figure 3 and the TFHE scheme [25, 26].

Signature Size. We first analyze the concrete size of the signature produced by the scheme in Section 4.3, which is exactly the signature size of the blind signature scheme. For efficiency, it is preferable to consider variants over polynomial rings (note that the above proofs carry over directly to the ring-SIS and module-SIS settings). We consider the DILITHIUM-G signature scheme from [32, Appendix B]. It is a variant of DILITHIUM [31], a concrete implementation of Lyubashevsky's signature scheme designed to reach a compromise between efficiency and simplicity. DILITHIUM-G has slightly smaller signature sizes than DILITHIUM, but requires that $\mathbf{y}$ has Gaussian coordinates rather than having them uniform in an interval. This makes it somewhat more cumbersome to implement, but the Rényi divergence analysis above is more effective than its uniform counterpart. Note that for the blind signature application described in Section 4.1, the sampling of $\mathbf{y}$ is done in the clear by the signer.

Rather than defining new parameter sets for DILITHIUM-G (which would most likely lead to improved concrete parameters), we show that DILITHIUM-G already offers significant security when removing rejection sampling. We consider two variants of DILITHIUM-G: with the Bai-Galbraith [11] signature truncation technique as in [32], and without it.[1] Indeed, a variant without the truncation technique may be more convenient to evaluate homomorphically, depending on the specific choice of HE scheme.

For security, we place ourselves in the (classical/quantum) core-SVP hardness model, like in [32]. State of the art cryptanalysis suggests that bit security is expected to be at least 10-15 bits higher than core-SVP security (see [73, Section 5.2]). We consider the 'medium', 'recommended' and 'very high' parameter sets from [32]. For a given value of Q_S, and use the analysis above to derive core-SVP hardness estimates for this number of signature queries. Concretely, we use Equations (4.1) and (4.2) with $\delta \approx 1$ (which can be assumed in the multi-user setting, and can also be justified by considering known attack strategies [32]) and $a \approx 1$ (which minimizes the upper bound on the Rényi divergence). By adapting the figures from [32, Table 2], we obtain Table 2.

For example, using the 'very high' parameter set, we lose no more than 100 bits of security by removing rejection sampling, when the number of signature queries is limited to 256. In this setting, the overall classical bit security remains close to 128 bits (more precisely, 118 bits of classical core-SVP hardness), for a signature size that is below 3kB.

On the Choices of Signature and HE *scheme.* We use the TFHE scheme [25, 26]. With this choice, the relation proved by π_{sk} is linear in short unknowns, for which specific lattice-based techniques are known (see [35, 56], among others). Note that TFHE does not satisfy malicious circuit privacy, as required in the proof of Theorem 3. We upgrade it to a maliciously circuit private HE as described in Section 4.5. For our specific application to the blind signature from Subsection 4.1, the modification to circuit privacy induces a bootstrap for every encrypted bit prior to any computation, and $\Omega(\lambda)$ bootstraps for every encrypted bit at

[1] Concretely, Algorithm 14 from [32] is modified by removing Step 6, replacing $\mathbf{w_1}$ by $\mathbf{w}$ in Step 7, removing Step 13 and replacing $\mathbf{h}$ by $\mathbf{z_2}$ in Step 14. Algorithm 15 is updated accordingly.

	medium	recommended	very high
Standard deviation σ of $\mathbf{z}$	48127	44868	31082
Dimension m of $\mathbf{z}$	1536	2048	2560
Secret key magnitude bound d	6	5	3
Hamming weight α of $\mathbf{c}$	60	60	60
pk size	1184	1568	1952
sig size with truncation	1850	2435	2950
sig size without truncation	3200	4225	5105
Security with rejection	109 / 99	162 / 147	228 / 206
Security loss with $Q_S = 64$	25	27	25
Security loss with $Q_S = 128$	50	53	50
Security loss with $Q_S = 256$	100	106	100

Table 2: Security of DILITHIUM-G without rejection sampling. Sizes are given in bytes, security corresponds to classical/quantum core-SVP hardness, security losses are in bits. Note that due to a different Gaussian normalization, our Gaussian standard deviation is $\sqrt{2\pi}$ times higher than in [32, Table 2]. Adapting the methodology from [32, Appendix B.5], we bound the no-truncation signature size by $32+(2.25+\log_2(\sigma/\sqrt{2\pi}))\cdot m/8$ bytes.

the end of the computation. As explained in [34], the precise number of required bootstrappings depends on how much noise a ciphertext can accommodate. As a loose estimate, we will set this number to 40. For the signature scheme, we start from DILITHIUM-G, but we consider a variant in which the 256-dimensional challenge $\mathbf{c}$ is binary uniform rather than ternary of weight 60. This is because the conversion between the output of a cryptographic hash function to this specific format seems complex to perform homomorphically. This should negatively impact the figures in Table 2, in a moderate manner. For the sake of simplicity, we keep the figures from Table 2. We also assume that the message μ to be signed has bit-length 256, as the user can hash μ to a 256 bit-string before encrypting it to request a signature.

User Complexity and Transcript Sizes. The user must first create an HE key-pair $(\mathsf{pk}, \mathsf{sk})$ and a NIZKAoK π_{sk} of well-formedness of the HE key-pair. The public key includes key-switching keys and bootstrapping keys, which makes pk quite large: a back-of-the-envelope calculation leads to an estimate of 50MB. This is without π_{sk}. Even though this is quite heavy, it is worth pointing out that this data is independent from the message μ to be signed, and the computation of pk and π_{sk} and their transmission to the signer can be amortized across several executions of the blind signature protocol (including with different signers) and also be performed in an off-line phase.

From the perspective of the user, the computations related to the message μ to be signed are limited to an HE encryption (to produce ct_μ from μ) and an HE decryption (to recover σ from ct_σ). If using a Ring-LWE ciphertext in dimension $n = 1024$ with a 25-bit modulus, the bit-size of ct_μ is ~ 3.2kB. As this is a fresh encryption, we assumed for this computation that the first ring element of the ciphertext is set as the output of a random oracle. On the way back, the signature has bit-length below 3kB. This can be packed into 24 Ring-LWE ciphertexts, leading to a ct_σ with below 160kB. Both encryption and decryption are very efficient.

Signer Complexity. The signer samples $\mathbf{y} \leftarrow \mathcal{D}_{\mathbf{Z}^m,\sigma}$, sets $\mathbf{c} = H(\mathbf{A}\mathbf{y}, \mu)$ and $\mathbf{z} = \mathbf{y} + \mathbf{S}\mathbf{c}$ and outputs $\sigma = (\mathbf{z}, \mathbf{c})$. Note that $\mathbf{A}, \mathbf{y}$ and $\mathbf{S}$ are in clear, whereas μ is provided through ct_μ. Due to the non-algebraic nature of the computation of the hash function, we consider HE computations on bits, using the TFHE gate bootstraps (GBS). This requires to extract and keyswitch when receiving ct_μ and to pack to build ct_σ (as ct_μ is not very large, one may also consider sending a bit-by-bit encryption of ct_μ, to decrease the size of pk).

For circuit privacy, we bootstrap all ciphertexts after they have been converted to a bit-by-bit format. This requires 256GBS (one GBS for each bit of μ). As a GBS costs ~ 10ms, we estimate that this step can be performed in less than 3s.

For the hash function, we consider SHA3-256. It works with iteratively evaluating a permutation, consuming 1088 bits at every iteration. From an efficiency perspective, it is important to note that we can first perform iterations in the clear to consume $\mathbf{Ay}$, and as μ has 256 bits, only one iteration needs to be performed homomorphically to consume the rest of the input. Before the iteration, the message is xored with the current SHA3 state, which costs 256GBS. Now, one SHA3 iteration consists in 24 rounds, and each round requires $\sim 6,000$ binary gates, and hence $\sim 6,000$GBS when performed homomorphically. We estimate that this can be performed in less than $\sim 1,500$s if done sequentially. Note that the computations within a round can be massively parallelized.

As $\mathbf{c}$ is obtained in a bit-by-bit encrypted form, we suggest to perform the computation of $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$ using a bit-level addition. A back-of-the-envelope computation suggests that this should take below 1 million GBS, i.e., $\sim 10,000$s if done sequentially. Note that this can also be massively parallelized.

Finally, to guarantee circuit privacy, we use 40GBS for each output bit. We estimate the cost to $\sim 1,200$s, if done sequentially. Again, this can be parallelized.

4.5 Maliciously Circuit Private HE

We consider the GSW HE scheme [42]. In our instantiation of the HE-based blind signature, we consider TFHE [25, 26], which can be viewed as a variant of the GSW HE scheme with a very efficient bootstrapping procedure. To make it semi-honest circuit private, we use the construction of [34], which essentially consists in adding bootstrappings at the end of the computation. The alternative construction of [21] would require to modify TFHE more significantly.

The secret key sk and public key pk are, respectively:

$$\mathbf{s} = (-\bar{\mathbf{s}}, 1) \quad \text{and} \quad \hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \bar{\mathbf{s}}^\top \mathbf{A} + \mathbf{e}^\top \end{pmatrix},$$

where $\bar{\mathbf{s}} \leftarrow \{0,1\}^{n-1}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{(n-1)\times m}$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha}$. Here $m = n\lceil \log_2 q \rceil$. Note that in GSW, the secret vector $\mathbf{s}$ is uniform modulo q, whereas it is binary in the GSW scheme.

We modify the above binary GSW scheme to make it maliciously circuit private, as follows:

1. The matrix $\mathbf{A}$ is generated using a hash function modeled as a random oracle, to guarantee that it is uniformly distributed. This does not have any negative impact on efficiency.

2. The public key pk contains a NIZKAoK that the last row of $\hat{\mathbf{A}}$ is of the form $\mathbf{x}^\top \mathbf{A} + \mathbf{y}^\top$ for some low-norm vector $(\mathbf{x}, \mathbf{y})$. Note that such an argument is already contained in the blind signature scheme from Subsection 4.1, so this addition is superfluous in this application scenario.

3. Before HE computations are performed, each ciphertext is bootstrapped. As in [1], this ensures that every (possibly malicious) ciphertext becomes a properly formed ciphertext.

4. At the end of the HE computations, each ciphertext is bootstrapped $\Omega(\lambda)$ times, as in [34].

Assume first that the public key is well-formed (this is the purpose of modifications 1 and 2). The addition of pre-computation bootstrappings ensures that the (refreshed) ciphertexts are well-formed. The technique from [34] only requires from the HE that the scheme enables bootstrapping, which is the case here as pk is well-formed. Using [34], we obtain that the genuine and simulated distributions match. Now, in the case in which the public key is possibly not well-formed, if an adversary can distinguish between the genuine and simulated distributions, it can be used to break the soundness of the NIZKAoK.

4.6 Using A Simpler Hash Function in Lyubashevsky's Signature

Below we provide details of the variant using a simpler hash function, adapted from [24]. The hash function $H : \{0,1\}^* \to \mathbb{Z}_q^n$ is modeled as a random oracle in the security proof. As its input is only μ, in our

$\mathsf{KeyGen}(1^\lambda)$: Upon input the security parameter λ, set $q, n, m, \beta, d, \sigma$ such $n = \Omega(\lambda)$ and the scheme is secure (see Theorem 9); then do the following:

1. Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{S} \leftarrow \{-d, \ldots, 0, \ldots, d\}^{m \times n \log q}$.
2. Set $\mathbf{T} = \mathbf{AS}$.
3. Output $\mathsf{vk} = (\mathbf{A}, \mathbf{T})$, $\mathsf{sk} = \mathbf{S}$.

$\mathsf{Sign}(\mathsf{sk}, \mu)$: Upon input the signing key sk and a message μ, do the following:

1. Sample $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$.
2. Set $\mathbf{c} = \mathbf{G}^{-1}(\mathbf{Ay} + H(\mu)) \in \{0,1\}^{n \log q}$.
3. Set $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$.
4. Output $(\mathbf{z}, \mathbf{c})$.

$\mathsf{Verify}(\mathsf{vk}, \mu, (\mathbf{z}, \mathbf{c}))$: Upon input the verification key vk, a message μ, and a signature $(\mathbf{z}, \mathbf{c})$, do the following:

1. Check $\mathbf{c}$ is binary and $\|\mathbf{z}\| \leq \gamma$, where $\gamma = (2\sigma + dn \log q)\sqrt{m}$.
2. Check if $[\mathbf{A} \| \mathbf{G} + \mathbf{T}] \begin{bmatrix} -\mathbf{z} \\ \mathbf{c} \end{bmatrix} = H(\mu)$.
3. If all checks pass, then accept, else reject.

Figure 4 The Chen et al. Signature Without Aborts

blind signature construction from Subsection 4.1, the value $H(\mu)$ can be computed in the clear by the user before being encrypted and sent to the signer: the signer does not have to homomorphically evaluate H.

The correctness of the scheme is proved as in [24] (the removal of aborts has no impact on the correctness proof). Next, we establish security via the following theorem.

Theorem 9. Assume that $\mathsf{SIS}_{q,n,m,\beta}$ is hard for $\beta = 2\gamma + 2dn \log q \sqrt{m}$ and that $\sigma \geq nd \log q \sqrt{mQ}$, where Q is the maximum number of signing queries an attacker can make. Then the construction in Figure 4 satisfies $\mathsf{UF\text{-}CMA}$ in the random oracle model.

Proof. We prove the security via the following hybrids:

Hybrid_0: This is the real world, i.e., with honest executions of the Sign algorithm on signing queries by the adversary.

Hybrid_1: In this hybrid the challenger responds to the signing query for any message μ as follows.

1. Sample $\mathbf{y} \leftarrow \mathcal{D}_{\mathbf{Z}^m, \sigma}$.
2. Sample $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ and let $\mathbf{c} = \mathbf{G}^{-1}(\mathbf{u})$.
3. Set $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$.
4. Set $H(\mu) = \mathbf{u} - \mathbf{Ay} = \mathbf{Gc} - \mathbf{Ay}$.
5. Output $(\mathbf{z}, \mathbf{c})$.

Hybrid_2: In this hybrid the challenger responds to the signing query for any message μ as follows.

1. Sample $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ and let $\mathbf{c} = \mathbf{G}^{-1}(\mathbf{u})$.
2. Sample $\mathbf{z} \leftarrow \mathcal{D}_{\mathbf{Z}^m, \sigma}$.
3. Set $H(\mu) = \mathbf{Gc} - \mathbf{Ay}$.

4. Output $(\mathbf{z}, \mathbf{c})$.

The only difference between Hybrid_0 and Hybrid_1 is that in Hybrid_1, $H(\mu)$ is programmed to $\mathbf{u} - \mathbf{A}\mathbf{y}$, where $\mathbf{u}$ and $\mathbf{y}$ are chosen randomly. Hence, $\mathbf{u} - \mathbf{A}\mathbf{y}$ is indistinguishable from a random and hence, by the definition of the random oracle, Hybrid_0 and Hybrid_1 are indistinguishable As in Lemma 2, the following result holds.

Claim 10. If there is an adversary who makes at most Q_S signing queries and can win the game in Hybrid_2 with probability δ, then the probability of winning in Hybrid_3 is polynomial in δ, if $\sigma \geq nd \log q \sqrt{mQ_S}$.

Finally, it follows from [24, Theorem 4.16] that if there is a forger $\mathcal{F}$ who succeeds in forging a valid signature with non-negligible probability in Hybrid_2, then we can define an algorithm $\mathcal{B}$ which given $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, finds a non-zero $\mathbf{v}$ such that $\|\mathbf{v}\| \leq (2\gamma + 2dn \log q \sqrt{m})$ and $\mathbf{A}\mathbf{v} = \mathbf{0}$ with non-negligible probability. □

Efficiency Analysis. We now consider the cost inferred by using this variant of Lyubashevsky's signature inside the HE-based blind signature construction of Subsection 4.1. This involves the homomorphic evaluation of the signing algorithm, in which μ is given as an HE ciphertext and the signing key is provided in clear. As before, we consider the TFHE scheme.

At first, replacing SHA3-256 by the bit-decomposition function may seem to lead to a significant improvement. Note first that here, the user can send an encryption of $H(\mu)$ rather than μ, hence avoiding an HE evaluation of H. The efficiency improvement would come from the fact that SHA3-256 involves a large number of binary gates whereas, at the binary level, the bit-decomposition function is vacuous. A more precise analysis suggests that, in fact, this change negatively impacts efficiency. Nevertheless, we include this variant since a careful implementation with additional tricks may improve current estimates.

1. First, in the SHA3-256 approach of Subsections 4.3 and 4.4, the user was only sending a 256-bit encrypted message to the signer, leading to a ciphertext of $\sim$ 3kB. Now, the user needs to provide an encryption of $H(\mu)$, which is uniformly distributed in the range of $\mathbf{A}\mathbf{y}$. A back-of-the-envelope calculation suggests that the size of the first transcript grows to $\sim$ 100kB. This is of the same order of magnitude as the size of the second transcript of the blind signature scheme.

2. Second, the bit-level HE-computation of $\mathbf{A}\mathbf{y} + H(\mu)$ mod q requires a large number of gates. If computing over $\mathbb{Z}_q^n$ with $n = 1024$ and $\log_2 q \approx 25$, we estimate the number of gates to be of the order of 50,000. Using gate bootstrappings (GBS) that cost $\sim 10ms$ each, this leads to a cost of $\sim$ 500s (this can be parallelized). This is indeed a priori more efficient than the corresponding step of Subsection 4.4.

3. However, in Subsection 4.4, the most costly step is the computation of $\mathbf{z} = \mathbf{y} + \mathbf{S}\mathbf{c}$. Unfortunately, the bit-decomposition approach makes this step significantly more costly, as it requires $\mathbf{c}$ to be of dimension $\log_2 q \approx 25$ times bigger. The cost may not grow by a factor 25 (as the computations are made on integers of unbalanced bit-sizes).

Bit-level HE computations as provided by TFHE may not the best strategy here, as the $\mathbf{A}\mathbf{y} + H(\mu)$ and $\mathbf{y} + \mathbf{S}\mathbf{c}$ computations are linear. However, if using another HE scheme, the computation of the bit-decomposition function may become more difficult. One would also need to assess how to adapt the other ingredients of the blind signature scheme, such as the $\mathsf{NIZKAoK}$ argument π_{sk} of well-formedness of the HE key-pair.

5 Two Round Blind Signature from One-More-ISIS

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

5.1 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 5.6.

Informally, the one-more-ISIS assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell+1$ GPV signatures [41], 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 5.1. 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 $\mathbf{C}\mathbf{y}_j = \mathbf{t}_j$, $\|\mathbf{y}_j\| \le \beta$ 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 [14]. We could define a variant of one-more-ISIS inspired from the known target version of the one-more-RSA inversion problem from [14], in which the set S is restricted to be of size $\ell+1$. The choice (chosen target) formulation from Definition 5.1 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.

5.2 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 NIZKAoK for the statement of Equation (5.1) (see Figure 5).
3. A CPA-secure PKE scheme PKE(PKE.KeyGen, PKE.Enc, PKE.Dec) that is perfectly correct.

The construction is provided in Figure 5. The parameters q, n, m, σ are set such that Lemma 1 is applicable, the distribution of $\mathbf{A}\mathbf{x}$ is close to uniform at Step 1 of the signing algorithm (using Lemmas 4 and 5 with standard deviation parameter $\sigma/m = \Omega(1)$), and one-more-ISIS$_{q,m,n,\sigma,2\beta}$ is hard with $\beta = 2\sigma\sqrt{m}$.

5.3 Completeness of the blind signature scheme

The completeness of the scheme follows from the choice of β, correctness of $\mathsf{SamplePre}$ and the completeness of the argument system $\mathsf{NIZKAoK}$. In more detail, we make the following observations to argue completeness. From the correctness of $\mathsf{SamplePre}$, the vector $\mathbf{y}$ is small and satisfies $\mathbf{Cy} = \mathbf{t}$, where $\mathbf{t} = \mathbf{Ax} + H(\mu)$. This gives us $\mathbf{Cy} - \mathbf{Ax} = H(\mu)$. Furthermore, the vector $\mathbf{x}$ is small by design and $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x}\|\mathbf{y}; r)$ by construction. Hence, the proof π for Equation (5.1) verifies and the user accepts the proof because of the completeness of $\mathsf{NIZKAoK}$.

We now make a few remarks about the construction. Observe that we choose $\mathbf{x}$ to have norm at most β/m, which is a factor m smaller than that of $\mathbf{y}$. This is because in the security proof, we will construct solutions to the $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ problem as $\mathbf{y} - \mathbf{Rx}$ (see Step 5 of the unforgeability proof), where $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$. Thus, choosing $\|\mathbf{x}\| \leq \beta/m$ and $\|\mathbf{y}\| \leq \beta$ allows us to bound the norm of the $\mathsf{one\text{-}more\text{-}ISIS}$ solution by 2β as desired. Note that by increasing the ratio between the norms of $\mathbf{x}$ and $\mathbf{y}$ further, one can decrease the quantity 2β to a value that is arbitrarily close to β (hence possibly weakening the hardness assumption). Another important component is the inclusion of ciphertext $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x}\|\mathbf{y}; r)$ in the signature. It enables to circumvent rewinding in the extraction of all the witnesses $(\mathbf{x}_i\|\mathbf{y}_i)$ of the $Q_S + 1$ message-signature pairs output by the adversary, in the proof of unforgeability (see Step 5). Without it, the reduction would need to rewind $Q_S + 1$ times to extract all the pairs $(\mathbf{x}_i, \mathbf{y}_i)$, to construct the $\mathsf{one\text{-}more\text{-}ISIS}$ solution, leading to a security loss exponential in Q_S.

5.4 Security

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

Theorem 11. Assume that the $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ assumption holds and the argument $\mathsf{NIZKAoK}$ is sound. Then the blind signature scheme in Figure 5 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 $\mathsf{PKE.sk}$. For every signature $\sigma_j = (\pi_j, \mathsf{ct}_j)$ output by the adversary (for $j \in [\ell + 1]$), it uses $\mathsf{PKE.sk}$ to decrypt ct_j into a plaintext μ_j (which can be $\perp$ in case decryption fails). It stores the μ_j's.

Hybrid_2 : This hybrid differs from the previous one in the way matrix $\mathbf{A}$ is chosen. The challenger first samples a binary matrix $\mathbf{R} \leftarrow \{0,1\}^{m\times m}$ and sets $\mathbf{A} = \mathbf{CR}$.

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. The only difference between Hybrid_1 and Hybrid_2 is that in the latter $\mathbf{A}$ is computed as $\mathbf{CR}$, where $\mathbf{R}$ is a uniform binary matrix, instead of sampling it uniformly randomly from $\mathbb{Z}_q^{n\times m}$. The two hybrids are indistinguishable because Lemma 4 implies that $(\mathbf{C}, \mathbf{A})$ is within statistical distance $2^{-\Omega(\lambda)}$ from $(\mathbf{C}, \mathbf{CR})$.

Claim 1. Assume that the NIZK argument system is sound and PKE is correct. Then 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_2, then there exists a $\mathsf{one\text{-}more\text{-}ISIS}$ adversary $\mathcal{B}$ with Q_S preimage queries.

Note that this claim implies the result.

Proof. The reduction is as follows.

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 $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ 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 $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(n, m, q)$.
- Sample $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$.
- Output $\mathsf{BSig.sk} = \mathbf{T_C}$, $\mathsf{BSig.vk} = (\mathbf{C}, \mathbf{A}, \mathsf{PKE.pk})$.

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 $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma/m}$.
   - It computes $\mathbf{t} = \mathbf{A}\mathbf{x} + H(\mu)$.
   - 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 $\mathbf{C}\mathbf{y} = \mathbf{t}$.
   - It sends $\mathbf{y}$ to the user.
3. **User:** Upon receiving $\mathbf{y}$, user $\mathcal{U}$ does the following:
   - It verifies that $\|\mathbf{y}\| \leq \beta$ and satisfies $\mathbf{C}\mathbf{y} = \mathbf{t}$.
   - It samples $\mathsf{PKE.Enc}$ randomness r and computes $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r)$.
   - It generates a $\mathsf{NIZKAoK}$ π for following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathbf{A}, \mathsf{PKE.pk})$, ct and μ, there exists r and vectors $\mathbf{x}, \mathbf{y}$ such that

$$\|\mathbf{x}\| \leq \beta/m \ \wedge \ \|\mathbf{y}\| \leq \beta \wedge \ \mathbf{C}\mathbf{y} - \mathbf{A}\mathbf{x} = H(\mu) \\ \wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r). \tag{5.1}$$

   - The signature is (π, ct).

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

Figure 5 Blind Signature from $\mathsf{one\text{-}more\text{-}ISIS}$.

1. Upon being challenged by the $\mathsf{one\text{-}more\text{-}ISIS}$ challenger $\mathcal{C}$, with matrix $\mathbf{C}$, algorithm $\mathcal{B}$ does the following:
   - It samples a uniform binary matrix $\mathbf{R}$ and computes $\mathbf{A} = \mathbf{C}\mathbf{R}$.
   - It samples $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
   - It invokes $\mathcal{A}$ with $(\mathbf{A}, \mathbf{C}, \mathsf{PKE.pk})$ 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 $\mathbf{C}\mathbf{y}' = \mathbf{z}$. 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 pairs of short vectors $(\mathbf{x}_j, \mathbf{y}_j)$. If all μ_j's have been hash-queried by $\mathcal{A}$ and the vectors $(\mathbf{x}_j, \mathbf{y}_j)$ satisfy Equation (5.1) for all $j \in [Q_S + 1]$, then $\mathcal{B}$ outputs $\{(\mathbf{y}_j - \mathbf{R}\mathbf{x}_j, H(\mu_j))\}_{j \in [Q_S+1]}$. If any decryption fails or any of the above conditions are not satisfied, $\mathcal{B}$ aborts.

Note that $\mathcal{A}$'s view is identical to the one in Hybrid_2. 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 $\mathsf{NIZKAoK}$, 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 (μ_j) is equal to $\mathbf{C}\mathbf{y}_j - \mathbf{A}\mathbf{x}_j$. Finally, by the soundness of $\mathsf{NIZKAoK}$, the following indeed holds for all $j \in [Q_S+1]$:

$$\|\mathbf{x}_j\| \leq \beta/n \ \wedge \ \|\mathbf{y}_j\| \leq \beta \ \wedge \ \mathbf{C}\mathbf{y}_j - \mathbf{A}\mathbf{x}_j = H(\mu_j).$$

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 $j \in [Q_S+1]$,

$$\mathbf{t}_j = \mathbf{C}\mathbf{y}_j - \mathbf{A}\mathbf{x}_j = \mathbf{C}\mathbf{y}_j - \mathbf{C}\mathbf{R}\mathbf{x}_j = \mathbf{C}(\mathbf{y}_j - \mathbf{R}\mathbf{x}_j).$$

Since $\mathbf{R}$ is a binary matrix, we have $\|\mathbf{y}_j - \mathbf{R}\mathbf{x}_j\| \leq 2\beta$ for all $j \in [Q_S+1]$.

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 $\mathcal{C}$. □

Next we show that the construction satisfies honest signer blindness.

Theorem 12. Assume that PKE is $\mathsf{IND\text{-}CPA}$ secure and the $\mathsf{NIZKAoK}$ is zero-knowledge. Then the blind signature in Figure 5 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 genuinely computing the $\mathsf{NIZKAoK}$s, the challenger simulates them without using the witnesses.

Hybrid_2 : This hybrid differs from the previous hybrid in that both ct_0 and ct_1 encrypt $\mathbf{0}$ instead of $(\mathbf{x}_0\|\mathbf{y}_0)$ and $(\mathbf{x}_1\|\mathbf{y}_1)$, 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 $\mathbf{t}_0 = \mathbf{A}\mathbf{x}_0 + H(\mu_b)$ (resp. $\mathbf{t}_1 = \mathbf{A}\mathbf{x}_1 + H(\mu_{\bar{b}})$), 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 $\mathsf{NIZKAoK}$.
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 4), 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. Similarly to the construction in Section 3, 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. By choosing a suitable encryption scheme so that $\mathsf{PKE.pk}$ is computationally indistinguishable to uniform, one can set $\mathsf{PKE.pk}$ as the output of a random oracle on a publicly-known value.

5.5 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 NIZKAoK (see Figure 5). 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 [69] (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 NIZKAoK and instead rely on algebraic proofs for linear relations [35, 56]. This lets us reduce the signature size to maybe as small as 50kB [35, 56], 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 [38] 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).

5.6 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 5.1 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. [41]. 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 5), 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 6, 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.

The correctness of Algorithm 6 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 $\|\mathbf{y}\| < 2\sqrt{\lfloor \frac{n}{w} \rfloor \cdot m} \cdot \sigma =$

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 $\mathbf{y} = T[\mathbf{a}_{i_1}] + \ldots + T[\mathbf{a}_{i_{\lceil n/w \rceil}}]$.

Figure 6 Combinatorial Attack on one-more-ISIS.

$\Theta(\sqrt{1 + \frac{n \log q}{\log(Q/n^2)}} \cdot \sqrt{m} \cdot \sigma)$, 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 [50, 41]. 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 [50, 41] instantiated with the basis $\mathbf{B}$ and the syndrome vector $\mathbf{t}$ (as in Lemma 1). 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 [69, 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 $\beta = \Theta(m\sigma)$.

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., [13]). 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 [45, 72] 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, 60, 13], 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 [30]. 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, Ilaria Chillotti, Léo Ducas, 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, and BPI-France in the context of the national project RISQ (P141580). 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] S. Agrawal, S. Goldwasser, and S. Mossel. Deniable fully homomorphic encryption from learning with errors. In *CRYPTO*, 2021.

[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] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. *J. Cryptol.*, 2018.

[13] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, 2016.

[14] 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.

[15] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT*, 2019.

[16] F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)security of ROS. In *EUROCRYPT*, 2021.

[17] O. Blazy, P. Gaborit, J. Schrek, and N. Sendrier. A code-based blind signature. In *ISIT*, 2017.

[18] D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *International Workshop on Public Key Cryptography*, pages 1–16. Springer, 2011.

[19] J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO*, 2019.

[20] 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.

[21] F. Bourse, R. del Pino, M. Minelli, and H. Wee. FHE circuit privacy almost for free. In *CRYPTO*, 2016.

[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] Y. Chen, A. Lombardi, F. Ma, and W. Quach. Does Fiat-Shamir require a cryptographic hash function? In *EUROCRYPT*, 2021.

[25] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT*, 2016.

[26] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In *ASIACRYPT*, 2017.

[27] N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *ASIACRYPT*, 2001.

[28] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO*, 2012.

[29] 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.

[30] E. Doulgerakis, T. Laarhoven, and B. de Weger. Finding closest lattice vectors using approximate Voronoi cells. In *Post-Quantum Cryptography*, 2019.

[31] 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.

[32] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS – Dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, 2017. Version 1, dated 27/06/2017.

[33] L. Ducas and T. Prest. Fast Fourier orthogonalization. In *ISSAC*, 2016.

[34] L. Ducas and D. Stehlé. Sanitization of FHE ciphertexts. In *EUROCRYPT*, 2016.

[35] M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT*, 2020.

[36] 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.

[37] M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *CRYPTO*, 2006.

[38] 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/.

[39] S. Garg and D. Gupta. Efficient round optimal blind signatures. In *EUROCRYPT*, 2014.

[40] S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signatures. In *CRYPTO*, 2011.

[41] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[42] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, 2013.

[43] 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.

[44] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, 2012.

[45] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, 2011.

[46] E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-based blind signatures, revisited. In *CRYPTO*, 2020.

[47] J. Howe, T. Prest, T. Ricosset, and M. Rossi. Isochronous gaussian sampling: From inception to implementation. In *PQCrypto*, 2020.

[48] S. Ibrahim, M. Kamat, M. Salleh, and S. A. Aziz. Secure E-voting with blind signature. In *NCTT*, 2003.

[49] A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO*, 1997.

[50] P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, 2000.

[51] A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *EUROCRYPT*, 2014.

[52] H. Q. Le, W. Susilo, T. X. Khuc, M. K. Bui, and D. H. Duong. A blind signature from module lattices. In *DSC*, 2019.

[53] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, 2013.

[54] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, 2009.

[55] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

[56] V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *PKC*, 2021.

[57] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, 2010.

[58] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

[59] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 2007.

[60] P. Q. Nguyên and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2008.

[61] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, 1992.

[62] D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. *IET Information Security*, 2019.

[63] A. Petzoldt, A. Szepieniec, and M. S. E. Mohamed. A practical multivariate blind signature scheme. In *Financial Crypto*, 2017.

[64] D. Pointcheval and J. Stern. Provably secure blind signature schemes. In *ASIACRYPT*, 1996.

[65] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT*, 1996.

[66] D. Pointcheval and J. Stern. New blind signatures equivalent to factorization (extended abstract). In *CCS*, 1997.

[67] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptol.*, 2000.

[68] T. Pornin and T. Prest. More efficient algorithms for the NTRU key generation using the field norm. In *PKC*, 2019.

[69] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 2009.

[70] M. Rückert. Lattice-based blind signatures. In *ASIACRYPT*, 2010.

[71] C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptol.*, 1991.

[72] C.-P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, 1994.

[73] P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, G. Seiler, and D. Stehlé. CRYSTALS-Kyber: Algorithm specifications and supporting documentation (version 3.0). https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.

[74] J. Stern. A new paradigm for public key identification. *IEEE Trans. Inf. Theory*, 1996.

[75] K. Takashima and A. Takayasu. Tighter security for efficient lattice cryptography via the rényi divergence of optimized orders. In *ProvSec*, 2015.

[76] 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.

[77] X. Yi and K.-Y. Lam. A new blind ECDSA scheme for bitcoin transaction anonymity. In *Asia-CCS*, 2019.

[78] C. Yin, S. Huang, P. Su, and C. Gao. Secure routing for large-scale wireless sensor networks. In *ICCT*, 2003.

[79] R. K. Zhao, R. Steinfeld, and A. Sakzad. COSAC: compact and scalable arbitrary-centered discrete Gaussian sampling over integers. In *PQCrypto*, 2020.