Practical, Round-Optimal Lattice-Based Blind Signatures

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

Abstract. Blind signatures are a fundamental cryptographic primitive with numerous practical applications. While there exist many practical blind signatures from number-theoretic assumptions, the situation is far less satisfactory from post-quantum assumptions. In this work, we provide the first overall practical, lattice-based blind signature, supporting an unbounded number of signature queries and additionally enjoying optimal round complexity. We provide a detailed estimate of parameters achieved – we obtain a signature of size slightly above 45KB, for a core-SVP hardness of 109 bits. The run-times of the signer, user and verifier are also very small.
Our scheme relies on the Gentry, Peikert and Vaikuntanathan 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, called the one-more-ISIS assumption. This assumption can be seen as a lattice analogue of the one-more-RSA assumption by Bellare *et al.* [JoC'03]. To gain confidence in our assumption, we provide a detailed analysis of diverse attack strategies.

1 Introduction

[1] Blind signatures are a fundamental cryptographic primitive with numerous applications in e-cash [26], e-voting [49], cryptocurrencies [79] and many others. In a blind signature scheme [26], a user $\mathcal{U}$, holding a public key and message, may request a signature from a signer $\mathcal{S}$, holding a signing key, such that the signer is not able to link a message-signature pair with a protocol execution, and the user is not able to forge signatures even after multiple interactions with the signer.

Blind signatures have been studied for several decades, and admit instantiations from a variety of assumptions [27, 71, 40, 51, 41, 42, 38, 57]. Given their wide applicability, there has been a significant thrust towards obtaining practical efficiency. Constructions based on standard assumptions are primarily feasibility results [42, 38] which do not admit practical instantiations. In light of this, in the number-theoretic regime, reasonable new assumptions were introduced to obtain efficient constructions. For instance, in the group setting, several candidates [27, 68, 71, 47, 40] are based on the hardness of the non-standard ROS/mROS problem (note that the ROS problem was recently broken [17]) or rely [1, 77] on the algebraic group [51] and the generic group [67] models, which are very strong idealizations. The situation is analogous in the regime of pairings [21, 19, 41] or RSA [15].

Post-Quantum Regime. Under post-quantum assumptions, the situation is much more unsatisfactory – even disregarding efficiency, several lattice-based blind signatures [73, 6, 5, 23, 54, 69] were found to have errors in their security proofs [48]. The recent construction by Hauck *et al.* [48] aimed to fix the errors but the resulting construction is completely impractical – using their suggested parameters, the constructed blind signature has size ≈ 7.73MB, for security against adversaries limited to getting 7 signatures. The very recent work of Lyubashevsky *et al.* [57] achieves better parameters (signature size of about 150KB), but the cost of their signing algorithm grows linearly in the maximum number of signatures that an adversary can query. This makes it impractical for situations where the number of signatures is large or cannot be apriori bounded.

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

[1] Revised version dated October 10, 2026. The definition of one-more-ISIS (Definition 4.1) now requires the targets $\mathbf{t}_j$ to be distinct (see Remark 4.2), and the proofs of Theorems 4.3 and A.1 have been revised accordingly.

Finally there are constructions based on codes [18] and systems of algebraic equations [70] but these are either impractical or do not satisfy the standard definition of security.

Subsequent to the public appearance of the present work, del Pino and Katsumata [29] also provided a two round lattice-based blind signature. Their techniques and final result are incomparable to ours – on one hand, their signature is of size 102.6KB, which is more than twice as large as ours, and their transcript size is 851KB, which is about 18 times as large as ours. On the other hand, their construction relies on the hardness of the standard MSIS and MLWE assumptions, while ours relies on a new hardness assumption called the one-more-ISIS assumption (described below). Additionally, they show how to upgrade their construction to be secure in the quantum random oracle model (albeit at the cost of making the transcript size 770 times larger than ours) while we do not consider this extension in the present work.

Our Results. In this work, we provide the first overall practical, lattice-based blind signature, which additionally enjoys optimal round complexity. Our scheme relies on the Gentry, Peikert and Vaikuntanathan (GPV) signature [43] 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, called one-more-ISIS. This assumption can be seen as a lattice analogue of the one-more-RSA assumption by Bellare *et al.* [JoC'03]. Informally, the one-more-ISIS assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell+1$ GPV signatures [43], even when given access to up to ℓ inversions of arbitrarily chosen syndromes.

Our construction supports an unbounded number of signatures and is overall more efficient than all prior candidates. While it is based on a new assumption, we believe that for a practice oriented primitive like blind signatures, it is justified to introduce plausible assumptions as was done in the number-theoretic regime. We provide detailed cryptanalysis to justify our new assumption.

1.1 Our Techniques

The starting point of our work is a two round blind signature by Fischlin [38], which relies on the CRS model. To begin, we adapt this scheme to the ROM and instantiate it with efficient lattice based signatures and non-interactive zero knowledge proofs (NIZK). Due to the extensive research in efficient lattice based signatures [43, 56, 36, 45, 39, 12, 32] and proof systems [55, 30, 16, 78, 22, 36, 34, 60] over the last 15 years, this already provides a candidate which is "somewhat reasonable" in practice.

Adapting Fischlin's Protocol. Our adaptation of Fischlin's protocol uses a public key encryption scheme PKE and a non-interactive zero knowledge argument of knowledge NIZKAoK as building blocks. To begin, we consider the honest signer model for blindness, in which it is assumed that the signing and verification keys are generated honestly, though the signer can deviate arbitrarily from the signing protocol. This assumption will subsequently be removed. We summarize this protocol next. In what follows, we assume some familiarity with the signature scheme of Gentry, Peikert and Vaikuntanathan (GPV); please refer to [43] for a refresher.

In the setup phase, we run $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and discard $\mathsf{PKE.sk}$. Next, following the GPV signature scheme, we sample a matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ of it. We set the signing key of the blind signature as $\mathsf{BSig.sk} = \mathbf{T_C}$, and the verification key as $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$.

To sign the message μ, the user $\mathcal{U}$ samples $\mathsf{PKE.Enc}$ randomness r and computes $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mu; r)$. It sends ct to the signer. Upon receiving ct, the signer $\mathcal{S}$ computes a GPV signature on ct and returns this to the user. In more detail, it computes $H(\mathsf{ct})$ and uses the trapdoor $\mathbf{T_C}$ to sample $\mathbf{y}$ such that $\mathbf{y}$ is short and $\mathbf{Cy} = H(\mathsf{ct})$ (modulo q). It sends $\mathbf{y}$ to the user. Here H is a hash function, modeled as a random oracle in the security proof.

Upon receiving $\mathbf{y}$, the user $\mathcal{U}$ verifies that $\mathbf{y}$ is small and that $\mathbf{Cy} = H(\mathsf{ct})$ and aborts if this fails. It generates a non-interactive zero-knowledge argument of knowledge (NIZKAoK) π for the following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$ and μ, there exists PKE randomness r and a vector $\mathbf{y}$ such that

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

In the above, β is some appropriate bound. Finally, the user outputs π as the signature. To verify the blind signature, the verifier checks that the proof π is valid. Thus, the final signature in the blind signature protocol is a NIZKAoK that the user knows a GPV signature for an *encryption* of the message.

For full-fledged blindness, it suffices to ensure that PKE.pk has been honestly generated by the adversarial signer, without a corresponding decryption key. This can be achieved, for example, by choosing PKE such that PKE.pk is computationally indistinguishable from uniform, and then setting PKE.pk as the output of another hash function H' modeled as a random oracle, on an arbitrary public input.

Since the witness of the NIZKAoK includes the randomness r used to compute the ciphertext, and the ciphertext is inside a (complex) hash function, the statement that we require to prove becomes very complex and resorting to general purpose NIZKAoK seems unavoidable. Despite amazing recent advances in efficient general purpose NIZKAoK [16, 9], the resulting parameters are formidable – as discussed in Section 3, we estimate a proof size of more than 100KB and prover time complexity of one hour or more. Even worse, the prover of the NIZKAoK is the user in the blind signature, who is generally expected to be computationally light. This leads to a blind signature with very large user time complexity, which is very dissatisfying, both in theory and practice.

An Efficient Construction from one-more-ISIS. We begin by observing that general purpose NIZKAoKs are the primary source of inefficiency in the above protocol, and "lightening" the usage of NIZKAoK would result in a significantly lighter overall protocol. Intuitively, some usage of NIZKs feels unavoidable if we want to stick to a two round protocol, but can we simplify the statement that is proved? 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 NIZK for linear statements with small unknowns, for which practical constructions have been developed recently [34, 60, 58]. Armed with these ideas, we provide a simple, overall efficient protocol as follows.

For setup, we run $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and discard PKE.sk. Again, discarding PKE.sk can be achieved in the real world by setting PKE.pk as the output of a hash function on a public value (this requires ensuring that the distributions match). Next, we sample a matrix $\mathbf{C}$ together with trapdoor $\mathbf{T_C}$ as before. At this stage, we depart from the previous protocol – instead of encrypting the message μ to achieve blindness, we will rely on a much simpler "one time pad" style blinding mechanism. For this, we sample another matrix $\mathbf{A}$ and set $\mathsf{BSig.sk} = \mathbf{T_C}$, $\mathsf{BSig.vk} = (\mathbf{C}, \mathbf{A}, \mathsf{PKE.pk})$. For full fledged blindness, we would also need to set $\mathbf{A}$ as the output of a random oracle, together with PKE.pk as discussed above.

For signing a message μ, a user $\mathcal{U}$ samples a vector $\mathbf{x}$ from a suitable distribution such that $\mathbf{Ax}$ is indistinguishable from uniform. It computes $\mathbf{t} = \mathbf{Ax} + H(\mu)$ and sends $\mathbf{t}$ to the signer. Note that for suitable choice of $\mathbf{x}$, the term $H(\mu)$ and hence μ is hidden from the view of the signer. Upon receiving $\mathbf{t}$, signer $\mathcal{S}$ uses the trapdoor $\mathbf{T_C}$ to sample a short vector $\mathbf{y}$ such that $\mathbf{Cy} = \mathbf{t}$ (modulo q). It sends $\mathbf{y}$ to the user. Upon receiving $\mathbf{y}$, user $\mathcal{U}$ verifies that $\mathbf{y}$ is short and satisfies $\mathbf{Cy} = \mathbf{t}$. It samples PKE.Enc randomness r and computes

$$\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r).$$

It generates a NIZK π for the following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathbf{A}, \mathsf{PKE.pk})$, ct and μ, there exist r and vectors $\mathbf{x}, \mathbf{y}$ such that

$$\|\mathbf{x}\| \leq \beta_1 \wedge \|\mathbf{y}\| \leq \beta_2 \quad \wedge \quad \mathbf{Cy} - \mathbf{Ax} = H(\mu)$$
$$\wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r).$$

In the above, β_1 and β_2 are appropriate parameters and H is the random oracle hash function. The signature is (π, ct), and verification consists in verifying the NIZK π as before.

Note that the above statement also involves the hash function H which is modeled as a random oracle in the security proof. But, crucially, the input μ to H is known, implying that $H(\mu)$ can be seen as a *public* quantity and this does not make the proof complex. By using Regev's encryption scheme [72] (or variants of it), one can ensure 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 can instead rely on

NIZK for linear relations with small unknowns [34, 60, 58]. This lets us reduce the signature size to 45.19KB, as against more than 100KB. More importantly, the cost of generating and verifying the proof becomes very small.

The astute reader may wonder why the witnesses $\mathbf{x}\|\mathbf{y}$ are being encrypted. In the unforgeability proof, this allows to circumvent rewinding when extracting GPV preimages from the output of the adversary. Rewinding would incur a loss that is exponential in the number of preimages that the attacker requested from the signer. Please see Section 4 for more details.

The resultant protocol is extremely simple and appears quite similar to the first protocol we presented, which in turn is a natural adaptation of Fischlin's protocol from 2006 [38]. The reader may wonder whether replacing the ciphertext computed by the user in the first step by a one time pad is the only difference from the first scheme, and if so, why efficient lattice-based blind signatures have remained elusive for so long. The key new insight of our work is in formulating a meaningful new assumption that allows reducing security of this very natural construction to it. We describe our assumption next and discuss how it implies security of our candidate.

The one-more-ISIS *Assumption.* 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 $\mathbf{C}\mathbf{y}_j = \mathbf{t}_j$, $\|\mathbf{y}_j\| \le \beta$ and the $\mathbf{t}_j$'s are distinct and 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 of the chosen target version of the one-more-RSA inversion problem from [15]. It is also closely related to the k-SIS problem [20] which was introduced in the context of linearly homomorphic signatures. The k-SIS problem is as follows: Given a matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$, and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \mathbb{Z}^m$ satisfying $\mathbf{A}\cdot\mathbf{e}_i = \mathbf{0} \bmod q$, find a short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{A}\cdot\mathbf{e} = \mathbf{0} \bmod q$, such that $\mathbf{e}$ is not in $\mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$. In [20], 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 it has already seen; this prevents it 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.

In particular even if the attacker manages to obtain a trapdoor for the matrix $\mathbf{C}$ via repeated preimage queries to the vector $\mathbf{0}$, this trapdoor will not be of sufficiently good quality to lead to an attack. In more detail, such a trapdoor enables sampling preimages to arbitrary images, and hence the attacker can output $\ell+1$ pairs of the form $\{(\mathbf{y}_j, \mathbf{t}_j)\}_{j\in[\ell+1]}$, such that $\mathbf{C}\mathbf{y}_j = \mathbf{t}_j$ and $\mathbf{t}_j$ were provided via syndrome queries, for all $j \in [\ell+1]$. However, it will be unable to meet the constraint that $\|\mathbf{y}_j\| \le \beta$. We believe this assumption is very natural and are optimistic that it may have other applications.

Given our new assumption, one more unforgeability follows very naturally. In the proof, the challenger can sample the PKE public and secret keys using the PKE setup algorithm, and not discard the secret key. Assuming correctness of PKE and with knowledge of PKE.sk, the challenger can extract the pairs $(\mathbf{x}_j, \mathbf{y}_j)$ corresponding to the signature of each message μ_j. We have by soundness of the NIZK that $\mathbf{C}\mathbf{y}_j - \mathbf{A}\mathbf{x}_j = H(\mu_j)$. By setting $\mathbf{A} = \mathbf{C}\cdot\mathbf{R}$ for a low norm matrix $\mathbf{R}$, we can (i) use the leftover hash lemma to argue that $\mathbf{A}$ appears uniform, and (ii) rewrite $\mathbf{C}\mathbf{y}_j - \mathbf{A}\mathbf{x}_j$ as $\mathbf{C}(\mathbf{y}_j - \mathbf{R}\mathbf{x}_j)$. Finally, by programming the random oracle so that $H(\mu_j)$ is a syndrome queried by the one-more-ISIS adversary yields the proof. Please see Section 4 for more details.

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. Please see Section 4.5 for more details.

Estimating Performance. We provide a detailed analysis of the performance of our new candidate in Section 4.4. To instantiate our new protocol based on one-more-ISIS, we use the following building blocks:

- For the hash function, we use SHA-3-256;
- For the trapdoor generation and preimage sampling, we follow Falcon-512 [39];
- For the IND-CPA secure PKE, we adapt the CRYSTALS-Kyber encryption scheme [10];
- For the NIZK scheme, we follow the recent protocol of [58, Figure 10].

To make these building blocks compatible with each other, we need a working modulus which must satisfy the following constraints:

- Its prime factors must be sufficiently large to avoid soundness improving repetitions in the zero knowledge proof;
- The moduli of the underlying signature and encryption schemes should divide the working modulus so that the relations required to be proven are simpler;
- The polynomial $x^{128} + 1$ defining the ring used in the NIZK scheme should split in exactly two prime factors modulo all factors of the working modulus, due to technical reasons related to the NIZK.

Satisfying the above constraints requires a delicate balancing of parameters, which results in a number of changes in the building blocks. First, we must modify Falcon's modulus because $x^{128} + 1$ splits completely modulo the original Falcon modulus, which violates the last constraint above. We must also instantiate the CRYSTALS-Kyber framework so that it complies with the zero-knowledge proof π from [58] and enjoys perfect correctness. Since it is the zero-knowledge proof that makes most of the overall signature size as well as the complexity to generate it, we are mainly interested in making the generation of π efficient, while potentially sacrificing the efficiency of the other routines.

We provide a detailed guideline on how to instantiate the NIZKAoK protocol from [58, Figure 10], and how to choose the parameters for the other building blocks so as to obtain a concrete estimate for all the parameters of the resultant blind signature scheme. We provide a Python script (included with the submission) that estimates the concrete security of the building blocks, as well as the size of the resulting signatures. The resulting protocol has security relying on Ring-LWE [75, 62], Module-LWE [24, 53] and the Module-SIS [53] variant of one-more-ISIS.

Using our script, we obtain a signature of size less than 44KB, for a classical core-SVP hardness of 109 bits (following the security methodology from [7]). Note that bit security is typically estimated to be higher than core-SVP hardness (see [10]), and we expect it to be of the order of 128 bits. The transcript has size less than 1.5KB. The costs of the signer and user in the signing protocol, as well as that of the verifier are also very low. To see this, note that the signer must simply compute a GPV pre-image, the user must compute a ciphertext and proof for a linear statement with small unknowns, while the verifier must verify this proof. Thus, in the end, we obtain a protocol which enjoys security under a post-quantum assumption and is overall more efficient than all prior candidates.

Other Related Works. Aside from lattice based blind signatures, there are a few other constructions from conjectured post-quantum assumptions. The most relevant to our work is the code-based construction of Blazy *et al.* [18], relying on the CFS signature scheme [28] and Stern zero-knowledge proofs [76]. Like in our one-more-ISIS construction, their construction relies on a new assumption, related to CFS. However, there are important differences with our work. In CFS, not all syndromes can be inverted, and the procedure needs to be repeated if no inversion is possible. Hence, the resulting blind signature scheme is not round optimal. Moreover, due to the poor scaling of CFS signatures and the use of Stern proofs, their construction achieves signature size of several MB. A blind signature based on multivariate polynomial systems was described in [70], with a non-standard unforgeability security property.

2 Preliminaries

Notation. We write vectors with bold small letters and matrices with bold capital letters. Let S be any set, then $|S|$ represents the cardinality of S, while in case of any $x \in \mathbb{R}$, $|x|$ represents the absolute value of x. For any $n \in \mathbb{N}$, we let the set $\{1, 2, \ldots, n\}$ be denoted by $[n]$. For a distribution D over a countable set $\mathcal{X}$, we let $H_\infty(D) = -\max_{x \in \mathcal{X}} \log_2 D(x)$ denote the min-entropy of D. The statistical distance between two distributions D_0 and D_1 over $\mathcal{X}$ is defined as $\frac{1}{2} \sum_{x \in \mathcal{X}} |D_0(x) - D_1(x)|$.

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

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

2.1 Blind Signatures

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

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

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

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

Blind signatures must satisfy two security properties: one more unforgeability and blindness [50].

Definition 2.2 (One More Unforgeability). *The blind signature* $BS = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *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* $(\mathsf{BSig.sk}, \mathsf{BSig.vk})$*, and algorithm* $\mathcal{U}^*$ *is given* $\mathsf{BSig.vk}$.
2. *Algorithm* $\mathcal{U}^*$ *interacts concurrently with* Q_S *instances* $\mathcal{S}^1_{\mathsf{BSig.sk}}, \ldots, \mathcal{S}^{Q_S}_{\mathsf{BSig.sk}}$.
3. *Algorithm* $\mathcal{U}^*$ *outputs* $(\mu_1, \sigma_1, \ldots, \mu_{Q_S+1}, \sigma_{Q_S+1})$.

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

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

Definition 2.3 (Honest Signer Blindness). *The blind signature* $BS = (\mathsf{Gen}, \mathcal{S}, \mathcal{U}, \mathsf{Vrfy})$ *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* $(\mathsf{BSig.sk}, \mathsf{BSig.vk})$ *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* $\mathcal{U}_0 := \mathcal{U}(\mathsf{BSig.vk}, \mu_b)$ *and* $\mathcal{U}_1 := \mathcal{U}(\mathsf{BSig.vk}, \mu_{\bar{b}})$ *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 $(\mathsf{BSig.sk}, \mathsf{BSig.vk})$ at Step 1 (possibly maliciously), and gives $\mathsf{BSig.vk}$ to the challenger.

2.2 Non-Interactive Zero Knowledge Arguments

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

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

The argument system Π *should satisfy the following properties.*

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

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

- ***Soundness:*** *Let* L *be the language corresponding to NP relation* R*. For any* $x \in \{0,1\}^{\mathrm{poly}(\lambda)}$ *such that* $x \notin L$ *and any* $2^{o(\lambda)}$ *time prover* P^**, we have*

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

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

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

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

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

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

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

2.3 Lattices and Discrete Gaussians

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

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

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 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 parameters $\sigma > 0$ and $\mathbf{c}$ is defined as:

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

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

2.4 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, 43, 64], whereas the second is derived from [52, 43, 25].

Lemma 2.6. *Let q, n, m be positive integers with $q \geq 2$ and $m \geq 6n\log_2 q$.*

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

There is a PPT algorithm $\mathsf{SamplePre}(\mathbf{A},\mathbf{T},\mathbf{u},\sigma)$, which takes as input the above pair $(\mathbf{A},\mathbf{T})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a sufficiently large $\sigma = \Omega(\sqrt{n\log q\log m})$ and outputs a vector $\mathbf{e}$ from $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$. Further, with probability $1 - 2^{-\Omega(n)}$, we have $\|\mathbf{e}\| \leq \sigma\sqrt{m}$.

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

2.5 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 [72, 25] when it is appropriately parameterized.

Definition 2.7 (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 2.8 (Short Integer Solution (SIS)). *Let q, n, m, β be functions of a parameter λ. An instance of the $\mathsf{SIS}_{q,n,m,\beta}$ problem is a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$. A solution to the problem is a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \beta$ and $\mathbf{A} \cdot \mathbf{v} = \mathbf{0} \bmod q$.*

Like LWE, the SIS problem is known to be at least as hard as certain lattice problems in the worst case [2, 65, 43], when it is appropriately parameterized. The same holds for the *inhomogeneous* version of the SIS problem stated below.

Definition 2.9 (Inhomogeneous Short Integer Solution (ISIS)). *Let q, n, m, β be functions of a parameter λ. An instance of the $\mathsf{ISIS}_{q,n,m,\beta}$ problem is a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{t} \leftarrow \mathbb{Z}_q^n$. A solution to the problem is a vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \beta$ and $\mathbf{A} \cdot \mathbf{v} = \mathbf{t} \mod q$.*

2.6 Other Useful Lemmas

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

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

are within statistical distance ε.

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

The following lemma is adapted from [56], which uses a different Gaussian normalization. In our uses of the third item, for simplicity, we will set $k = \sqrt{2/\pi}$, for which the probability upper bound is $\leq 2^{-m}$.

Lemma 2.11 (Adapted from [56, Lemma 4.4]). *The following hold.*

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

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 [38].

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 in the unforgeability proof.
2. A CPA-secure PKE scheme PKE 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 $n = \Omega(\lambda)$, Lemmas 2.6 and 2.10 are applicable, and $\mathsf{SIS}_{q,m,n,2\beta}$ is hard with $\beta = \sigma\sqrt{m}$. The completeness of the scheme follows from the choice of β (using the Gaussian tail bound from Lemma 2.11) and the completeness of the $\mathsf{NIZKAoK}$.

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

Setup. $\mathsf{Gen}(1^\lambda)$: Upon input the security parameter λ, define $n, m, q, \sigma, \beta = \sigma\sqrt{m}$ as functions of λ such that q is prime, $\mathsf{SIS}_{q,n,m,2\beta}$ 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 $\|\mathbf{y}\| \leq \beta$ and $\mathbf{Cy} = H(\mathsf{ct})$ and aborts if this fails.
   - It generates a $\mathsf{NIZKAoK}$ π for following statement: Given $\mathsf{BSig.vk} = (\mathbf{C}, \mathsf{PKE.pk})$ and μ, there exists r and a vector $\mathbf{y}$ such that
   $$\|\mathbf{y}\| \leq \beta \ \wedge \ \mathbf{Cy} = H(\mathsf{Enc}(\mathsf{PKE.pk}, \mu; r)). \tag{3.1}$$
   - The signature is π.

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

Fig. 1 Adaptation of Fischlin's Blind Signature.

3.2 Security

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

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

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

Hybrid_0: This is the genuine one more unforgeability experiment.

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

Hybrid_2: The difference between this hybrid and the previous one is in how the hash and sign queries are answered. On a fresh input c for a hash query, the challenger first samples $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$ and returns $H(c) = \mathbf{C}\mathbf{y}$. 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 Lemmas 2.6, 2.10 and 2.11, the views of the adversary in Hybrid_1 and Hybrid_2 are within statistical distance $(Q_S + Q_H) \cdot 2^{-\Omega(\lambda)}$ from one another, where Q_S is the number of signing queries and Q_H is the number of hash queries[2].

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 an arbitrary such message and $\pi^\star$ be its 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 Lemma 2.11, we have $\|\mathbf{y}\| \leq \beta = \sigma\sqrt{m}$ with probability $1 - 2^{-\Omega(\lambda)}$ and $\mathbf{y} = \mathbf{y}^\star$ with probability $2^{-\Omega(\lambda)}$. We conclude that $\mathbf{y} - \mathbf{y}^\star$ is non-zero, has norm $\leq 2\beta$ and satisfies $\mathbf{C}(\mathbf{y} - \mathbf{y}^\star) = \mathbf{0}$, providing a solution to the $\mathsf{SIS}_{q,n,m,2\beta}$ instance $\mathbf{C}$.

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

Proof. We argue blindness using the following hybrids.

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

Indistinguishability of hybrids

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

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

[2] We note here that $\mathsf{SamplePre}$ is assumed to be deterministic (see Section 2.4), without which the claim would not be true.

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

3.3 Efficiency Estimate

We consider the following instantiation of the building blocks.

- For PKE, we can take any lattice-based public-key encryption scheme. It is only required to be IND-CPA, but it must be perfectly correct. The latter property can typically be guaranteed by tail-cutting error distributions and increasing the working modulus sufficiently. Also, lattice-based encryption schemes typically 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 [39], which is an efficient instantiation of the $\mathsf{TrapGen\text{-}SamplePre}$ framework from [43]. With this choice, the second transcript will have size below 1KB. Also, that makes the signer particularly efficient – for instance, using Falcon [39], 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 sponge absorbing steps for reading the input and 7 sponge squeezing steps 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 [16]. Estimating a precise cost is difficult, but we do not expect a proof of size below 100KB. We also do not expect the prover runtime to be below 1 hour, whereas verifier runtime could be significantly lower. It could be beneficial to use hash functions designed to be compatible with all-purpose NIZKAoK, such as [8, 44].

4 Two Round Blind Signature from One-More-ISIS

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

4.1 The One-More-ISIS Assumption

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

Informally, the $\mathsf{one\text{-}more\text{-}ISIS}$ assumption states that for any polynomially bounded ℓ, it is difficult to forge $\ell + 1$ GPV signatures [43], 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, $\mathsf{one\text{-}more\text{-}ISIS}$ could possibly be easier to solve than it is to break the chosen-message security of the GPV signature scheme.

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

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

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 $2^{-\Omega(\lambda)}$.*

Remark 4.2. The targets $\mathbf{t}_j$ are required to be distinct at Step 4. Otherwise, the problem would be easy to solve, with only two preimage queries and one syndrome query. An attacker, receiving a uniform syndrome $\mathbf{t}$, queries for a preimage $\mathbf{y_t}$ of $\mathbf{t}$, and a preimage $\mathbf{y_0}$ of the zero vector. Then the three vectors $\mathbf{y_t}$, $\mathbf{y_t} + \mathbf{y_0}$, $\mathbf{y_t} - \mathbf{y_0}$ are all somewhat short preimages of $\mathbf{t}$.[3]

The definition is reminiscent to the chosen target version of the one-more-RSA inversion problem from [15]. We could define a variant of one-more-ISIS inspired from the known target version of the one-more-RSA inversion problem from [15], in which the set S is restricted to be of size $\ell + 1$. The choice (chosen target) of formulation made in Definition 4.1 is driven by the security proof of the blind signature scheme. In the RSA setting, the chosen and known target versions reduce to one another, but this seems difficult to adapt to the ISIS setting.

4.2 Construction

The construction uses the following building blocks:

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

The construction is provided in Figure 2. The parameters q, n, m, σ are set such that Lemmas 2.6 and 2.10 are applicable, the distribution of $\mathbf{Ax}$ is close to uniform at Step 1 of the signing algorithm (using Lemmas 2.10 and 2.11 with Gaussian parameter $\sigma/m = \Omega(1)$), and one-more-ISIS$_{q,n,m,\sigma,2\beta}$ is hard with $\beta = \sigma\sqrt{m}$.

Completeness We make the following observations to argue completeness. From the correctness of 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 (4.1) verifies and the user accepts the proof because of the completeness of NIZK.

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 one-more-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 one-more-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 may need to rewind $Q_S + 1$ times to extract all the pairs $(\mathbf{x}_i, \mathbf{y}_i)$, to construct the one-more-ISIS solution, leading to a security loss exponential in Q_S.

[3] We thank Niklas Siemer for pointing this out; see also [13, Section 2.5].

Setup. $\mathsf{Gen}(1^\lambda)$: Upon input the security parameter λ, define $n, m, q, \sigma, \beta = \sigma\sqrt{m}$ as functions of λ such that q is prime, $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ 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 NIZK 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
   $$\|\mathbf{x}\| \leq \beta/m \ \wedge \ \|\mathbf{y}\| \leq \beta \wedge \ \mathbf{C}\mathbf{y} - \mathbf{A}\mathbf{x} = H(\mu) \wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r). \tag{4.1}$$
   - The signature is (π, ct).

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

Fig. 2 Blind Signature from $\mathsf{one\text{-}more\text{-}ISIS}$.

4.3 Security

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

Theorem 4.3. *Assume that* NIZK *is sound. Then if there exists an adversary* $\mathcal{A}$ *in the random oracle model that issues* Q_S *signing queries and* Q_H *hash queries and outputs* $Q_S + 1$ *signatures with probability* δ*, then there exists an algorithm* $\mathcal{B}$ *that runs in essentially the same time as* $\mathcal{A}$ *and requests* Q_S *preimage queries and wins the* $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ *game with probability at least* $\delta - 2^{-\Omega(\lambda)} - (Q_S + 1)(2^{-\Omega(\lambda)} + q^{-n}) - Q_H^2 q^{-n}$.

Proof. We construct the proof 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 $j \in [Q_S + 1]$), it uses $\mathsf{PKE.sk}$ to decrypt ct_j into a plaintext $(\mathbf{x}_j \| \mathbf{y}_j)$ (which can be $\perp$ in case decryption fails). It stores the $(\mathbf{x}_j \| \mathbf{y}_j)$'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 $\mathbf{A} = \mathbf{C}\mathbf{R}$.

Indistinguishability of hybrids

In the following, we let $\mathsf{Adv}_i^{\mathsf{omuf}}$ represent the advantage of $\mathcal{A}$ in the one more unforgeability game in Hybrid_i. Then $\mathsf{Adv}_0^{\mathsf{omuf}} = \delta$.

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. Thus $\mathsf{Adv}_1^{\mathsf{omuf}} = \mathsf{Adv}_0^{\mathsf{omuf}} = \delta$.

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 Lemmas 2.10 and 2.11 imply that $(\mathbf{C}, \mathbf{A})$ is within statistical distance $2^{-\Omega(\lambda)}$ from $(\mathbf{C}, \mathbf{CR})$. Thus $\mathsf{Adv}_2^{\mathsf{omuf}} \geq \mathsf{Adv}_1^{\mathsf{omuf}} - 2^{-\Omega(\lambda)} = \delta - 2^{-\Omega(\lambda)}$.

We conclude with the following claim.

Claim 4.4 *Assume that the* NIZK *argument system is sound. Then if there is an adversary $\mathcal{A}$ in the random oracle model that makes at most Q_H hash queries and Q_S signing queries, and succeeds in generating Q_S+1 signatures with probability ε in* Hybrid_2*, then there exists a* one-more-ISIS *adversary $\mathcal{B}$, with essentially the same run time as $\mathcal{A}$, with Q_S preimage queries with success probability at least $\varepsilon-(Q_S+1)(2^{-\Omega(\lambda)}+q^{-n})-Q_H^2q^{-n}$.*

Proof. The reduction $\mathcal{B}$ is as follows.

1. Upon being challenged by the one-more-ISIS challenger $\mathcal{C}$, with matrix $\mathbf{C}$, algorithm $\mathcal{B}$ does the following:
   - It uniformly samples a binary matrix $\mathbf{R}$ and sets $\mathbf{A} = \mathbf{CR}$.
   - 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 $\mathbf{Cy}' = \mathbf{t}'$. Algorithm $\mathcal{B}$ forwards $\mathbf{y}'$ to $\mathcal{A}$.
4. Eventually, adversary $\mathcal{A}$ outputs Q_S+1 message-signature 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 (4.1) for all $j \in [Q_S+1]$, then $\mathcal{B}$ outputs $\{(\mathbf{y}_j - \mathbf{Rx}_j, H(\mu_j))\}_{j\in[Q_S+1]}$. If any decryption fails or any of the above conditions is not satisfied, then $\mathcal{B}$ aborts.

First note that by the soundness of NIZK, the probability that a statement with a valid proof is false is bounded above by $2^{-\Omega(\lambda)}$. Now, since the ciphertext ct is part of the statement, we have by perfect correctness of PKE that it decrypts to the correct value. Hence, the overall probability that a decryption fails is $\leq (Q_S+1)\cdot 2^{-\Omega(\lambda)}$. Next, we claim that for each μ_j, adversary $\mathcal{A}$ must have issued a corresponding hash query to $\mathcal{B}$. This is because otherwise, there is only a q^{-n} probability that a fresh $H(\mu_j)$ is equal to $\mathbf{Cy}_j - \mathbf{Ax}_j$. Additionally, by the soundness of NIZK, it holds that for all $j \in [Q_S+1]$:

$$\|\mathbf{x}_j\| \leq \beta/m \ \wedge \ \|\mathbf{y}_j\| \leq \beta \ \wedge \ \mathbf{Cy}_j - \mathbf{Ax}_j = H(\mu_j).$$

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 $j \in [Q_S+1]$,

$$\mathbf{t}_j = \mathbf{Cy}_j - \mathbf{Ax}_j = \mathbf{Cy}_j - \mathbf{CRx}_j = \mathbf{C}(\mathbf{y}_j - \mathbf{Rx}_j).$$

Since $\mathbf{R}$ is a binary matrix, we have $\|\mathbf{y}_j - \mathbf{Rx}_j\| \leq 2\beta$ for all j. Finally, since all μ_j's are distinct as per Definition 2.2 and $\mathcal{B}$ answers each fresh hash query with a fresh syndrome query, the $\mathbf{t}_j$'s are distinct except with probability at most $Q_H^2q^{-n}$. Thus, the success probability of $\mathcal{B}$ is at least $\varepsilon-(Q_S+1)(2^{-\Omega(\lambda)}+q^{-n})-Q_H^2q^{-n}$.

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 $\mathcal{C}$. Hence $\mathcal{B}$ is a valid adversary in the one-more-ISIS game.

Next we show that the construction satisfies honest signer blindness.

Theorem 4.5. *Assume that* NIZK *is zero-knowledge. Then if there exists a signer $\mathcal{S}^*$ in the random oracle model that wins the honest signer blindness game for the blind signature scheme in Figure 2 with advantage δ, then there exists an adversary $\mathcal{B}$, with essentially the same runtime as $\mathcal{S}^*$, that wins the* IND-CPA *security game for* PKE *with advantage at least $\delta/2 - 2^{-\Omega(\lambda)}$.*

Proof. We argue blindness using the 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 NIZKs, 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

In the following, we let $\mathsf{Adv}_i^{\mathsf{bl}}$ represent the advantage of $\mathcal{S}^*$ in the honest signer blindness game in Hybrid_i. Then $\mathsf{Adv}_0^{\mathsf{bl}} = \delta$.

1. The only difference between Hybrid_0 and Hybrid_1 is in the way π_0 and π_1 are computed. The two hybrids are indistinguishable because of the zero-knowledge property of the NIZK. Thus $\mathsf{Adv}_1^{\mathsf{bl}} \geq \mathsf{Adv}_0^{\mathsf{bl}} - 2^{-\Omega(\lambda)} = \delta - 2^{-\Omega(\lambda)}$.
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 computationally indistinguishable because of the IND-CPA security of PKE. In particular, if the advantage of $\mathcal{S}^*$ in the honest signer blindness game in Hybrid_2 is $\mathsf{Adv}_2^{\mathsf{bl}}$, then there exists an adversary $\mathcal{B}$ against IND-CPA security of PKE with advantage $\mathsf{Adv}_{\mathsf{IND\text{-}CPA}}$ such that $2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} \geq \mathsf{Adv}_1^{\mathsf{bl}} - \mathsf{Adv}_2^{\mathsf{bl}} \geq \delta - 2^{-\Omega(\lambda)} - \mathsf{Adv}_2^{\mathsf{bl}}$. (Here, we consider twice of $\mathsf{Adv}_{\mathsf{IND\text{-}CPA}}$ since both ct_0 and ct_1 are replaced with encryptions of $\mathbf{0}$ and hence the IND-CPA security of PKE is called twice.).
3. The only difference between Hybrid_2 and Hybrid_3 is in the choice of the masking term for $H(\mu)$. Since the vectors $\mathbf{x}_0$ and $\mathbf{x}_1$ are only used in the computations of the vectors $\mathbf{t}_0$ and $\mathbf{t}_1$, we have by the leftover hash lemma[4] (Lemmas 2.10 and 2.11), that $\mathbf{A}\mathbf{x}_0$ and $\mathbf{A}\mathbf{x}_1$ are statistically indistinguishable from uniform $\mathbf{u}_0$ and $\mathbf{u}_1$. Hence, Hybrid_2 and Hybrid_3 are statistically indistinguishable. More concretely, we have $\mathsf{Adv}_3^{\mathsf{bl}} \geq \mathsf{Adv}_2^{\mathsf{bl}} - 2^{-\Omega(\lambda)} \geq \delta - 2^{-\Omega(\lambda)} - 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} - 2^{-\Omega(\lambda)}$.

However, in Hybrid_3, the adversary $\mathcal{S}^*$ has zero advantage in guessing the bit b since it is information theoretically hidden. Hence $\delta - 2^{-\Omega(\lambda)} - 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} - 2^{-\Omega(\lambda)} \leq 0$, which is equivalent to $\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} \geq \delta/2 - 2^{-\Omega(\lambda)}$.

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 and that the matrix $\mathbf{A}$ is uniform. 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. To ensure $\mathbf{A}$ is uniform, it can similarly be set as the output of a random oracle on a publicly known value. Please see Appendix A for more details.

4.4 Concrete Instantiation

The goal of this section is to describe a concrete instantiation of the scheme from Figure 2, and analyze the size of the resulting signature. We rely on the following building blocks:

– for the hash function, we use SHA-3-256;
– for the trapdoor generation $\mathsf{TrapGen}$ and preimage sampling $\mathsf{SamplePre}$ algorithms, we follow Falcon-512 [39];

[4] We observe that in place of LHL, we can also use LWE in this step, by letting $\mathbf{A} = (\mathbf{A}'\|\mathbf{I})$ and $\mathbf{x}^\intercal = (\mathbf{x}'^\intercal\|\mathbf{e}^\intercal)$. This would change statistical closeness to computational indistinguishability. We use this variant in the concrete instantiation described below.

– for the IND-CPA secure PKE, we use an instantiation of the scheme from [61] under the Module-LWE assumption, which may be viewed as a simplification of CRYSTALS-Kyber [10];
– for the NIZK scheme, we follow the protocol from [58, Figure 10].

Since it is the NIZK scheme that makes most of the signature size as well as the cost to generate it, we are mainly interested in making the generation of the proof π efficient, while potentially sacrificing the efficiency of the other components.

Choosing the moduli. For compatibility with [59], the modulus of the ZK proof is chosen as a product of primes that are congruent to 5 modulo 8. Further, we require these primes to be above $2^{12.8}$, to avoid too many soundness-boosting repetitions. The smallest such prime is 7213. Concretely, we set the preimage sampling modulus to $q_{\mathrm{F}} = 7213$, the PKE modulus to $q_{\mathsf{PKE}} = 7213^2$ and the ZK modulus to $q_{\mathrm{zk}} = 7213^2 \cdot 123637$. We chose q_{zk} as a multiple of q_{F} and q_{PKE} to simplify the linear relations to be proven (see [58, Section 6.3]).

Trapdoor generation and preimage sampling. We instantiate Falcon-512 over the ring $\mathcal{R}_{512} = \mathbf{Z}[x]/(x^{512}+1)$, where the computations are taken modulo prime $q_{\mathrm{F}} = 7213$. It allows us to build our TrapGen algorithm as [39, Algorithm 5] that generates an NTRU secret key as the trapdoor, and to use Klein's sampler [52] (also known as the GPV sampler [43]) for our SamplePre algorithm. Our modulus q_{F} slightly differs from the one proposed in [39], since the zero-knowledge proof construction we use requires $x^{128}+1$ to have only few (two in our case) factors modulo q_{F}. Note that our modulus is a little smaller than Falcon's (12289): as discussed in [37], moduli in this range have limited impact on the security. Also, this modulus change does not significantly impact the efficiency of the Falcon-512 preimage sampler [39, Algorithm 10], as the modulus plays a limited role in it.

The TrapGen routine generates an NTRU secret key $\mathbf{f}, \mathbf{g} \in \mathcal{R}_{512}$, with coefficients of each polynomial $\mathbf{f}$ and $\mathbf{g}$ taken from $\mathcal{D}_{\mathbb{Z},1.17\sqrt{q_{\mathrm{F}}/(2\cdot 512)}}$ (see [39, Algorithm 5]), and builds up a short basis for the corresponding NTRU lattice (as in [39, Algorithm 5]). The public key is $\mathbf{h} = \mathbf{g}/\mathbf{f} \bmod q_{\mathrm{F}}$, defining the NTRU lattice $\{\mathbf{y} = (\mathbf{y}_1 \| \mathbf{y}_2) \in \mathcal{R}_{512}^2 : \mathbf{h}\cdot\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{0} \bmod q_{\mathrm{F}}\}$. The short basis enables a SamplePre routine that, given on input $\mathbf{t}$, outputs a preimage $\mathbf{y} = (\mathbf{y}_1 \| \mathbf{y}_2)$ such that $\|\mathbf{y}\| < 1.1 \cdot \sqrt{2\cdot 512}\cdot\sigma_{\mathrm{F}}$, where $\sigma_{\mathrm{F}} = \frac{1.17}{\pi}\sqrt{q_{\mathrm{F}} \cdot \log(4\cdot 512\cdot(1+\sqrt{128\cdot 2^{64}}))/2}$ (see [39, Eq. (2.13-2.14)]). For this instantiation, the relation "$\mathbf{C}\mathbf{y} = \mathbf{t}$" from Figure 2 translates into

$$\mathbf{h}\cdot\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{t} \bmod q_{\mathrm{F}}. \tag{4.2}$$

Another minor difference with Falcon-512 is that we perform rejection sampling to guarantee that $\mathbf{y}_1$ has infinity norm below a prescribed bound. Concretely, this bound is set to $\lceil 4.15 \cdot \sigma_{\mathrm{F}} \rceil$ so that this acceptance probability is ≥ 0.52. This is to ensure that $\mathbf{y}_1$ always belongs to the plaintext space of the encryption scheme described below.

With all the above, we now have concrete values for the parameters n, m, q, β: $n = 512$, $m = 1024$, $q = q_{\mathrm{F}}$, $\beta = 1.1 \cdot \sqrt{2\cdot 512}\cdot\sigma_{\mathrm{F}}$. Going forward, the transcript of the blind signature scheme will consist of $\mathbf{t}$ and $\mathbf{y}_1$ (note that $\mathbf{y}_2$ can be recovered as $\mathbf{t} - \mathbf{h}\cdot\mathbf{y}_1$). Using the figures above, we obtain a transcript size of 1.37KB.

Blinding the message. We now explain how we instantiate Step (1) of the signing protocol in Figure 2. We sample $\mathbf{h}' \in \mathcal{R}_{512}$ uniformly modulo q_{F}: the vector $(\mathbf{h}'\|1)$ plays the role of $\mathbf{A}$ from Figure 2. We then choose $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{R}_{512}$ with coefficients bounded in ℓ_∞-norm and set

$$\mathbf{t} = \mathbf{h}'\mathbf{x}_1 + \mathbf{x}_2 + H(\mu) \bmod q_{\mathrm{F}}. \tag{4.3}$$

In particular, we choose $\|(\mathbf{x}_1\|\mathbf{x}_2)\|_\infty \leq 2$, and use the Ring-LWE assumption to argue the computational indistinguishability of $\mathbf{t}$ from uniform (as opposed to a statistical argument as in the proof of Theorem 4.5).

Important for our construction is the ability to transform mod-q_{F} linear relations defined over the ring $\mathcal{R}_{512}$ to mod-q_{F} linear relations defined over $\mathcal{R}_{128}$. Following [59, Section 2.8] we can map one linear relation from $\mathcal{R}_{512}$ to 4 linear relations from $\mathcal{R}_{128}$, thus Eqs. (4.2) and (4.3) can both be viewed as 4 relations over $\mathcal{R}_{128}$. This will become relevant in the zero-knowledge proof.

IND-CPA *secure* PKE We use $\mathcal{R}_{128} = \mathbf{Z}[x]/(x^{128}+1)$ as underlying ring, by compatibility with the proof system (though we could have kept Kyber's $\mathbf{Z}[x]/(x^{256}+1)$ and viewed it as an extension of $\mathcal{R}_{128}$). We let $\mathcal{S}_\tau$ denote the set of elements from $\mathcal{R}_{128}$ with ℓ_∞-norm $\leq \tau$. The rank of the plaintext space (12) is 3 times the Falcon dimension, as we will encrypt $\mathbf{m} = (\mathbf{y}_1 \| \mathbf{x}_1 \| \mathbf{x}_2)$: note that we do not encrypt $\mathbf{y}_2$ as it can be recovered from $\mathbf{m}$ and $H(\mu)$ by using Eqs. (4.2) and (4.3). The ℓ_∞-norm bound on all small variables ($\tau = 3$) and the Module-LWE rank (8) are set to obtain a sufficiently high hardness.

In Figure 3, all computations are performed modulo $q_{\mathsf{PKE}} = 7213^2$, which is set high enough to guarantee correctness. Note that we rely neither on ciphertext compression nor on the binomial distribution as in CRYSTALS-Kyber, for the sake of simplicity. With these parameters, the ciphertext occupies 8.01KB.

Setup
- τ: ℓ_∞-norm bound on all short elements in the scheme
- q_{PKE}: a modulus
- $p < q_{\mathsf{PKE}}$: a positive integer

KeyGen()
- Sample $\mathbf{A}_1 \in \mathcal{R}_{128}^{8\times 8}$ uniform modulo q_{PKE}
- Sample $\mathbf{S}_1, \mathbf{S}_2 \leftarrow U(\mathcal{S}_\tau^{12\times 8})$
- Compute $\mathbf{A}_2 = \mathbf{S}_1 \cdot \mathbf{A}_1 + \mathbf{S}_2$
- Set $\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2)$ and $\mathsf{sk} = \mathbf{S}_1$

Enc($\mathsf{pk} = (\mathbf{A}_1, \mathbf{A}_2)$, $\mathbf{m} \in \mathcal{R}_{128}^{12}$)
- Sample $\mathbf{s}, \mathbf{e}_1 \leftarrow U(\mathcal{S}_\tau^8)$, and $\mathbf{e}_2 \leftarrow U(\mathcal{S}_\tau^{12})$
- Compute $\mathbf{c}_1 = \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1$
- Compute $\mathbf{c}_2 = \mathbf{A}_2 \cdot \mathbf{s} + \mathbf{e}_2 + p \cdot \mathbf{m}$
- Return $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2)$

Dec($\mathsf{sk} = \mathbf{S}_1$, $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2)$)
- Compute $\mathbf{t} = \mathbf{c}_2 - \mathbf{S}_1 \cdot \mathbf{c}_1$
- Return $(\mathbf{t} - \mathbf{t} \bmod p)/p$

Fig. 3 Instantiation of PKE for Figure 2

The correctness follows from the fact that for a properly formed ciphertext $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2)$, we have $\mathbf{t} = \mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1 + p\mathbf{m} \bmod q_{\mathsf{PKE}}$. For well-chosen parameters, this is $< q_{\mathsf{PKE}}/2$, and we recover $\mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1 + p\mathbf{m} \bmod q_{\mathsf{PKE}}$ over the integers. To recover $\mathbf{m}$, it suffices to take the quotient modulo p (provided that $\mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1$ is sufficiently small). Overall, for the decryption to be (perfectly) correct we require that

(I) $\|\mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1 + p\mathbf{m}\|_\infty < q_{\mathsf{PKE}}/2$, so that $\mathbf{c}_2 - \mathbf{S}_1 \cdot \mathbf{c}_1$ is not scrambled in the first step of the decryption algorithm;
(II) $\|\mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1\|_\infty < p/2$, so that $\mathbf{S}_2 \cdot \mathbf{s} + \mathbf{e}_2 - \mathbf{S}_1 \cdot \mathbf{e}_1$ is not scrambled in the second step of the decryption algorithm.

These requirements should hold for all $\mathbf{S}_1, \mathbf{S}_2$ sampled during key generation, and for all $\mathbf{s}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{m}$ as small as guaranteed by the zero-knowledge proof. Note that the latter is more demanding than requesting it for $\mathbf{s}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{m}$ as small as honestly generated, because the proof is for the ℓ_2-norm (rather than ℓ_∞-norm) and it batches several norm bounds together to reduce the number of proved norm bounds, at the expense of a constant factor increase in norm bound. The above conditions are satisfied for $p = 49126$.

Zero-knowledge proof We instantiate the zero-knowledge proof using [58, Figure 10]. We need to prove knowledge of $\mathbf{y} = (\mathbf{y}_1 \| \mathbf{y}_2)$ and $\mathbf{x} = (\mathbf{x}_1 \| \mathbf{x}_2)$ with $\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{R}_{512} \cong \mathcal{R}_{128}^4$ with small norms, such that (combining Eqs. (4.2) and (4.3)):

$$\mathbf{h} \cdot \mathbf{y}_1 - \mathbf{h}' \cdot \mathbf{x}_1 + \mathbf{y}_2 - \mathbf{x}_2 = H(\mu) \bmod q_{\mathrm{F}}. \tag{4.4}$$

We also need to prove the well-formedness of ct, i.e., the existence of $\mathbf{s}, \mathbf{e}_1 \in \mathcal{R}_{128}^8$ and $\mathbf{e}_2 \in \mathcal{R}_{128}^{12}$ that are small and satisfy the relations of the **Enc** algorithm from Figure 3 (modulo q_{PKE}) for the message $\mathbf{m} = (\mathbf{y}_1 \| \mathbf{x}_1 \| \mathbf{x}_2)$.

We commit to the vector $(\mathbf{y}_1 \| \mathbf{y}_2 \| \mathbf{x}_1 \| \mathbf{x}_2 \| \mathbf{s} \| \mathbf{e}_1 \| \mathbf{e}_2)$, prove two linear relations involving this vector (one coming from Eq. (4.4) and the other from the encryption), and prove three ℓ_2-norm bounds:

(I) $\|(\mathbf{y}_1 \| \mathbf{y}_2)\| \leq \beta$,
(II) $\|(\mathbf{x}_1 \| \mathbf{x}_2)\| \leq \sqrt{2 \cdot 512} \cdot 2$,
(III) $\|(\mathbf{s} \| \mathbf{e}_1 \| \mathbf{e}_2)\| \leq \tau \cdot \sqrt{(8+8+12) \cdot 128}$.

We shall not repeat the steps of the zero-knowledge proof from [58], but instead make a guideline on how to instantiate the protocol in [58, Figure 10]. The reader is advised to follow it using Table 1. Note that the two linear relations we need to prove incur negligible additional cost in terms of size, see [58, Figure 4].

variable	description	instantiation
ρ	# of quadratic eqs.	0
ρ_{eval}	# of evaluations with const. coeff. 0	0
v_e	# exact norm proofs	3
v_d	# non-exact norm proofs	0
$\mathbf{s}_1$	committed message in the Ajtai part	$(\mathbf{y} \| \mathbf{x} \| \mathbf{s} \| \mathbf{e}_1 \| \mathbf{e}_2)$
$\mathbf{m}$	committed message in the BDLOP part	$\emptyset$
$\mathbf{s}$	$(\mathbf{s}_1 \| \mathbf{m})$	$(\mathbf{y} \| \mathbf{x} \| \mathbf{s} \| \mathbf{e}_1 \| \mathbf{e}_2)$
$\mathbf{E}_1$	public matrix proving that $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leq \beta_1$	$\begin{pmatrix} \mathbf{Id}_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{Id}_4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
β_1	upper bound on $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\|$	β
$\mathbf{E}_2$	public matrix proving that $\|\mathbf{E}_2\mathbf{s} - \mathbf{v}_2\| \leq \beta_2$	$\begin{pmatrix} 0 & 0 & \mathbf{Id}_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{Id}_4 & 0 & 0 & 0 \end{pmatrix}$
β_2	upper bound on $\|\mathbf{E}_2\mathbf{s} - \mathbf{v}_2\|$	$\sqrt{2 \cdot 512} \cdot 2$
$\mathbf{E}_3$	public matrix proving that $\|\mathbf{E}_3\mathbf{s} - \mathbf{v}_3\| \leq \beta_3$	$\begin{pmatrix} 0 & 0 & 0 & 0 & \mathbf{Id}_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{Id}_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{Id}_{12} \end{pmatrix}$
β_3	upper bound on $\|\mathbf{E}_3\mathbf{s} - \mathbf{v}_3\|$	$\tau \cdot \sqrt{28 \cdot 128}$
$\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$	public vectors proving that $\|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\| \leq \beta_i$	$\mathbf{0}, \mathbf{0}, \mathbf{0}$
$\|\mathbf{x}\|$	norm of $((\beta_i^{(e)})^2 - \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\|^2)_i$	$\sqrt{3 \cdot 128}$
p_1, p_2, p_3	number of rows of $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$	8, 8, 28
$c^{(e)}$	$128 \cdot \sum_i (p_i + 1)$	6016
$\alpha^{(e)}$	upper bound on $\|(\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \ldots \|\mathbf{E}_3\mathbf{s} - \mathbf{v}_3\|\mathbf{x})\|$	5840

Table 1 Instantiation of the protocol from [58, Figure 10]. The left-most column 'variable' and the middle column 'description' refer to the notations from [58, Figure 10], the right-most column 'instantiation' refers to our notations.

For concrete parameter selection, we refer the reader to Table 2. We instantiate the variables κ, l, η, ν as in [58].

The parameters $\gamma_1, \gamma_2, \gamma_e$ are rejection sampling parameters. They are set so that the expected number of rejections before producing a valid signature is small. Using [58, Section 6.1], we obtain that the expected number is $\approx 2\exp(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2} + \frac{1}{2\gamma_e^2}) \approx 10.4$.

The parameter $m_1 = 4 \cdot 4 + 2 \cdot 8 + 12 + 3 = 47$ counts the length of the committed message as a vector over $\mathcal{R}_{128}$ (adding 3 as we have three norm equations). The parameters n and m_2 are chosen such that the Module-SIS and Module-LWE problems that underlie the zero-knowledge protocol are sufficiently hard.

Following the compression technique of [32], we can reduce the proof size by cutting low-order bits of the commitment. There are two variables responsible for this cut: γ and D. To choose these variables we follow the approach from [58, Section 6.1].

With these parameters, the proof π has size 37.18KB, and the overall signature (including π and ct) has size 45.19KB. (Recall the transcript has size 1.37KB.) This is for classical core-SVP hardness of 109 bits. Below, we give precise figures for our security estimates. These estimates as well as the sizes of signature components can be verified via a script available at https://gitlab.com/ElenaKirshanova/onemoresis_estimates. For concrete estimates of ModuleLWE and ModuleSIS assumptions we rely on the work from [10], and for the NTRU assumption security – on the work from [33]. To compute the proof size π, we adapt the strategy from [58, Section 6.1] to our setting with the exception that we take the estimates on the entropy of a discrete Gaussian variable from [35]. This choice is inline with the recent work [37, Section 5].

variable	value	variable	value	variable	value
q_{zk}	6432507821053	κ	2	n	11
l	2	λ	10	m_1	47
γ_1	10	η	59	m_2	34
γ_2	1.5	ν	1	ℓ	0
γ_e	5	D	16	γ	2^{24}

Table 2 Concrete parameter selection for the zero-knowledge protocol from [58, Figure 10]. The columns 'variable' refer to the notations from [58].

Security. Let us summarize our security assumptions and their corresponding bit security levels.

1. The security of Falcon's signature scheme relies on two assumptions:
   (I) Key recovery security relies on the NTRU assumption (i.e., it is hard to recover $\mathbf{f}, \mathbf{g}$ from $\mathbf{h} = \mathbf{g}/\mathbf{h}$). The estimator from [33] states that this has core-SVP hardness of 135 bits.
   (II) Forgery security relies on Module-SIS hardness. To estimate it, we use the Dilithium script [32], which states that this has core-SVP hardness of 129 bits.
2. Given $\mathbf{h}'$, we argue that $\mathbf{h}'\mathbf{x}_1 + \mathbf{x}_2$ hides $\mathbf{x}_1, \mathbf{x}_2$ under the Module-LWE assumption. Again, we use the script from [32], which states that this has core-SVP hardness of 122 bits.
3. The security of the encryption scheme (for both the secret key and the ciphertext) relies on the Module-LWE assumption. Here we reach core-SVP hardness of 120 bits.
4. The zero-knowledge proof relies on the Module-SIS and Module-LWE assumptions (technically, the construction of [58] relies on the so-called Extended-Module-LWE, whose hardness is conjectured to be the same as plain Module-LWE). For both Module-SIS and Module-LWE, we obtain 109 bits of core-SVP hardness.
5. Finally, we estimate the hardness of solving one-more-ISIS with the norm bound to be the norm of the extracted solution. For this, we assume that $\mathbf{h}'$ is set as $\mathbf{x}'_1 \cdot \mathbf{h} + \mathbf{x}'_2$ in the unforgeability proof instead of "$\mathbf{A} = \mathbf{CR}$" (see the proof of Theorem 4.3), using the same Module-LWE assumption as we did to argue computational indistinguishability from uniform of $\mathbf{x}_1 \cdot \mathbf{h}' + \mathbf{x}_2$, i.e., with $\|\mathbf{x}'_1\|_\infty, \|\mathbf{x}'_2\|_\infty \leq 2$. The extracted solution is $(\mathbf{y}_1 - \mathbf{x}_1\mathbf{x}'_1 \| \mathbf{y}_2 - \mathbf{x}_1\mathbf{x}'_2 - \mathbf{x}_2)$, which has ℓ_2-norm $\leq \sqrt{\beta^2 + 2^4 \cdot 512 \cdot 4 + 2^2 \cdot 512 \cdot 2}$. Having $\mathbf{h}$, the hardness of finding a preimage of such norm is again a Module-SIS instance, which we estimate to be at 109 bits of core-SVP hardness. The one-more-ISIS attacks described in the next section all have higher costs.

4.5 Security Analysis of One-More-ISIS

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

The hardness of the one-more-ISIS problem as stated in Definition 4.1 primarily depends on the precise relation between β, the upper bound on the norm of the vectors $\mathbf{y}_i$'s the adversary must output, and the dimensions m and n of the input matrix $\mathbf{C}$. We also assume that σ – the Gaussian parameter of the preimage queries – is of order $\Omega(\sqrt{m})$, which is what we would expect from an efficient sampler, e.g. [43]. Note that a significantly smaller standard deviation, e.g., of order $\mathcal{O}(1)$, would invalidate the hardness of the one-more-ISIS assumption as extremely short $\mathbf{y}$'s would enable an adversary to solve one-more-ISIS (see the discussion below). In this section we make the hardness of the one-more-ISIS problem explicit by describing the parameter regimes for which this problem can be solved in polynomial time, and for which, as far as we know, the problem is exponentially hard. We consider two approaches to solve one-more-ISIS: combinatorial attacks and lattice-based attacks.

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

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

The algorithm can be generalized to a larger set A. The generalization, presented in Figure 4, makes the attack less efficient, but reduces the bound on β. It is parametrized by Q, the upper bound on the number of the preimage queries the attacker can issue. This is also the assumed upper bound on the memory capacity of the attacker, since the attack requires that all the responses are stored.

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

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

Fig. 4 Combinatorial Attack on one-more-ISIS.

The correctness of the algorithm in Figure 4 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 at Step 2. Note that $|A| \leq n^2 q^w$: by definition of w, the algorithm makes $\leq Q$ queries. Finally, we can bound the norm of the output as $\|\mathbf{y}\| < 2\sqrt{\lfloor \frac{n}{w} \rfloor \cdot m} \cdot \sigma = \Theta\left(\sqrt{1 + \frac{n \log q}{\log(Q/n^2)}} \cdot \sqrt{m} \cdot \sigma\right)$, with probability greater than $1 - 2^{-\Omega(m)}$. The algorithm is correct for any $1 \leq w \leq n$ computed at 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 [52, 43]. This allows to solve one-more-ISIS in poly(m) time with $\beta = \Omega(m\sigma)$ using $O(m^2)$ preimage queries. More precisely, the attacker performs the following:

1. Given $\mathbf{C}$, compute a basis of $\Lambda_q^{\perp}(C)$.

2. Query the preimage ISIS oracle $\Theta(m^2)$ times for $\mathbf{t} = \mathbf{0}$. From the oracle's answers and from the basis of $\Lambda_q^\perp(C)$ constructed in the previous step, compute a basis $\mathbf{B}$ for $\Lambda_q^\perp(\mathbf{C}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{C}\mathbf{y} = \mathbf{0} \bmod q\}$ such that the norms of Gram-Schmidt orthogonalization $\widetilde{\mathbf{B}}$ of $\mathbf{B}$ are bounded from above.
3. Given an input $\mathbf{t} \in \mathbb{Z}_q^n$ find any $\mathbf{z} \in \mathbf{Z}^m$ such that $\mathbf{C}\mathbf{z} = \mathbf{t}$.
4. Run Babai's Nearest Plane algorithm [11] on input $(\mathbf{B}, \mathbf{z})$. Let $\mathbf{v} \in \Lambda_q^\perp(C)$ be the output. Return $\mathbf{z} - \mathbf{v}$ as a one-more-ISIS solution for $\mathbf{t}$.

Let us analyse the quality of the vectors returned by 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 set of m linearly independent vectors from $\Lambda_q^\perp(\mathbf{C})$ with at least constant probability bounded away from 0 (see [72, Corollary 3.16]). According to [20, Proposition 4.7.], these linearly independent vectors will be of norm bounded from above and below by $\Theta(\sqrt{m}\sigma)$. Out of this set we can efficiently extract m linearly independent vectors by checking which ones form a subspace of the desired rank. Using [63, Lemma 7.1] we can convert this set to a basis $\mathbf{B}$, such that $\|\widetilde{\mathbf{B}}\| < \sqrt{m}\sigma$. Finally, Babai's Nearest Plane algorithm in Step 4 outputs a vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{z}\| \leq \frac{1}{2}(\sum_{i\in[m]} \|\widetilde{\mathbf{b}}_i\|^2)^{1/2}$, where the right-hand side of the inequality is bounded from above by $m\sigma$ with probability greater than $1 - 2^{-\Omega(m)}$. Furthermore, the returned vector $\mathbf{e} = \mathbf{z} - \mathbf{v}$ satisfies $\mathbf{C}\mathbf{e} = \mathbf{C}\mathbf{z} - \mathbf{C}\mathbf{v} = \mathbf{t}$ as $\mathbf{C}\mathbf{v} = 0$, hence giving a one-more-ISIS solution for $\beta = O(m\sigma)$. As the vectors $\widetilde{\mathbf{b}}_i$ are already somewhat short thanks to the Gaussian tail bound, we do not expect a significant decay in their norms when converting them to a basis and/or applying a basis reduction algorithm, like LLL or BKZ, on $\mathbf{B}$. Hence, the norm of $\mathbf{e}$ is expected to be close to the above upper bound, resulting in the one-more-ISIS solution for $\beta = \Theta(m\sigma)$.

Another strategy to improve the above bounds on β at higher costs is to obtain a 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., [14]). 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 [46, 74] 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 $\beta = \Theta\left(\sqrt{1 + \frac{n \log q}{\log(Q/n^2)}} \cdot \sqrt{m}\sigma\right)$ 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 two 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, 66, 14], 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 [31]. 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.

Acknowledgments. The authors thank Olivier Blazy, Sébastien Canard, Dipayan Das, Raphaël del Pino, Carmit Hazay, Adeline Roux-Langlois and Muthuramakrishnan Venkitasubramaniam for helpful discussions. They thank Vadim Lyubashevsky and Ngoc Khanh Nguyen for insightful discussions on [58]. We also thank Rafael Del Pino and Niklas Siemer for pointing out several errors in previous versions of this document.

This work was partly supported by the DST "Swarnajayanti" fellowship, an IndoFrench CEFIPRA project, National Blockchain Project, European Union Horizon 2020 Research and Innovation Program Grant 780701, BPI-France in the context of the national project RISQ (P141580), and the ANR AMIRAL project (ANR-21-ASTR-0016). Elena Kirshanova is supported by the Young Russian Mathematics scholarship and by the Russian Science Foundation grant N 22-41-04411, https://rscf.ru/project/22-41-04411/. Part of the research corresponding to this work was conducted while the authors were visiting the Simons Institute for the Theory of Computing.

References

1. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In *EUROCRYPT*, 2001.
2. M. Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996.
3. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, 1999.
4. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, 2001.
5. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In *Financial Crypto*, 2020.
6. N. A. Alkadri, R. E. Bansarkhani, and J. Buchmann. On lattice-based interactive protocols: An approach with less or no aborts. In *ACISP*, 2020.
7. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In *USENIX Security*, 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. R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber: Algorithm specifications and supporting documentation, 2017. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Round-1-Submissions.
11. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem (shortened version). In *STACS*, 1985.
12. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, 2014.
13. F. Baldimtsi, J. Cheng, R. Goyal, and A. Yadav. Non-interactive blind signatures: Post-quantum and stronger security. In *ASIACRYPT*, 2024.
14. A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, 2016.
15. 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.
16. E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT*, 2019.
17. F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)security of ROS. In *EUROCRYPT*, 2021.
18. O. Blazy, P. Gaborit, J. Schrek, and N. Sendrier. A code-based blind signature. In *ISIT*, 2017.
19. A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In *PKC*, 2003.

20. D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *PKC*, 2011.
21. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *ASIACRYPT*, 2001.
22. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO*, 2019.
23. 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.
24. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012.
25. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.
26. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.
27. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO*, 1992.
28. N. T. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *ASIACRYPT*, 2001.
29. R. del Pino and S. Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In *CRYPTO*, 2022.
30. 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.
31. E. Doulgerakis, T. Laarhoven, and B. de Weger. Finding closest lattice vectors using approximate Voronoi cells. In *PQCrypto*, 2019.
32. 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.
33. L. Ducas and W. van Woerden. NTRU fatigue: How stretched is overstretched? In *ASIACRYPT*, 2021.
34. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT*, 2020.
35. M. F. Esgin, R. Steinfeld, D. Liu, and S. Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. *IACR Cryptol. ePrint Arch.*, 2022.
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. T. Espitau, M. Tibouchi, A. Wallet, and T. Yu. Shorter hash-and-sign lattice-based signatures. In *CRYPTO*, 2022.
38. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *CRYPTO*, 2006.
39. 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, 2017. Technical Report. Specification available at https://falcon-sign.info/.
40. G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In *EUROCRYPT*, 2020.
41. S. Garg and D. Gupta. Efficient round optimal blind signatures. In *EUROCRYPT*, 2014.
42. S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signatures. In *CRYPTO*, 2011.
43. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
44. 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.
45. T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, 2012.
46. G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, 2011.
47. E. Hauck, E. Kiltz, and J. Loss. A modular treatment of blind signatures from identification schemes. In *EUROCRYPT*, 2019.
48. E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-based blind signatures, revisited. In *CRYPTO*, 2020.
49. S. Ibrahim, M. Kamat, M. Salleh, and S. A. Aziz. Secure E-voting with blind signature. In *NCTT*, 2003.
50. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO*, 1997.
51. J. Kastner, J. Loss, and J. Xu. On pairing-free blind signature schemes in the algebraic group model. In *PKC*, 2022.
52. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, 2000.

53. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 2015.
54. H. Q. Le, W. Susilo, T. X. Khuc, M. K. Bui, and D. H. Duong. A blind signature from module lattices. In *DSC*, 2019.
55. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, 2013.
56. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.
57. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Efficient lattice-based blind signatures via gaussian one-time signatures. In *PKC*, 2022.
58. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In *CRYPTO*, 2022.
59. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *ASIACRYPT*, 2021.
60. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *PKC*, 2021.
61. V. Lyubashevsky, A. Palacio, and G. Segev. Public-key cryptographic primitives provably as secure as subset sum. In *TCC*, 2010.
62. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, 2010.
63. D. Micciancio and S. Goldwasser. *Complexity of lattice problems - a cryptographic perspective*. Springer, 2002.
64. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.
65. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 2007.
66. P. Q. Nguyên and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2008.
67. M. Ohkubo and M. Abe. Security of some three-move blind signature schemes reconsidered. In *SCIS*, 2003.
68. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, 1992.
69. D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. *IET Information Security*, 2019.
70. A. Petzoldt, A. Szepieniec, and M. S. E. Mohamed. A practical multivariate blind signature scheme. In *Financial Crypto*, 2017.
71. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptol.*, 2000.
72. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 2009.
73. M. Rückert. Lattice-based blind signatures. In *ASIACRYPT*, 2010.
74. C.-P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 1994.
75. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, 2009.
76. J. Stern. A new paradigm for public key identification. *IEEE Trans. Inf. Theory*, 1996.
77. S. Tessaro and C. Zhu. Short pairing-free blind signatures with exponential security. *IACR Cryptol. ePrint Arch.*, 2022.
78. 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.
79. X. Yi and K.-Y. Lam. A new blind ECDSA scheme for bitcoin transaction anonymity. In *Asia-CCS*, 2019.

Appendix

A Full-fledged Blindness

In this section we provide a construction of a blind signature with full-fledged blindness. The construction differs from the one in Figure 2 in the way the PKE encryption key PKE.pk is generated.

A.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 in the unforgeability proof.
2. A NIZK for the statement of Equation (A.1) (see Figure 5).
3. A perfectly correct IND-CPA secure public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ having the property that the public key generated by $\mathsf{PKE.KeyGen}$ is computationally indistinguishable from a uniformly sampled value from its range.
4. A hash function $\mathsf{H}_{\mathsf{PKE}} : \{\} \to \mathcal{K}_{\mathsf{PKE}}$ that will be modeled as random oracle in the unforgeability proof. Here, represents an empty bitstring and $\mathcal{K}_{\mathsf{PKE}}$ is the public key space of PKE.
5. A hash function $\mathsf{H}_{\mathsf{A}} : \{\} \to \mathbb{Z}_q^{n \times m}$ that will be modeled as random oracle in the unforgeability proof.

The construction is provided in Figure 5.

Parameters and Completeness The parameters setting and the completeness argument are the same as in Section 4.2.

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

- Set $\mathsf{PKE.pk} = \mathsf{H}_{\mathsf{PKE}}()$.
- Compute $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(n, m, q)$.
- Sample $\mathbf{A} = \mathsf{H}_{\mathsf{A}}()$.
- Output $\mathsf{BSig.sk} = \mathbf{T_C}$, $\mathsf{BSig.vk} = \mathbf{C}$.

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 computes $\mathbf{A} = \mathsf{H}_{\mathsf{A}}()$.
   - 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 computes $\mathsf{PKE.pk} = \mathsf{H}_{\mathsf{PKE}}()$.
   - 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 NIZK π for following statement[a]: Given $\mathbf{C}, \mathbf{A} = \mathsf{H}_{\mathsf{A}}(), \mathsf{PKE.pk} = \mathsf{H}_{\mathsf{PKE}}()$, ct and μ, there exists r and vectors $\mathbf{x}, \mathbf{y}$ such that
   $$\|\mathbf{x}\| \leq \beta/m \ \wedge \ \|\mathbf{y}\| \leq \beta \wedge \ \mathbf{C}\mathbf{y} - \mathbf{A}\mathbf{x} = H(\mu) \wedge \ \mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{PKE.pk}, \mathbf{x} \| \mathbf{y}; r). \tag{A.1}$$
   - The signature is (π, ct).

Verifying. The verifier computes $\mathsf{PKE.pk} = \mathsf{H}_{\mathsf{PKE}}()$ and $\mathbf{A} = \mathsf{H}_{\mathsf{A}}()$ and accepts if the proof π is valid, and rejects if it is not.

[a] Note that this is the same statement as in (4.1) in Section 4.2.

Fig. 5 Blind Signature with full-fledged blindness from $\mathsf{one\text{-}more\text{-}ISIS}$.

A.2 Security

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

Theorem A.1. *Assume that* NIZK *is sound. Then if there exists an adversary $\mathcal{A}$ in the random oracle model that issues Q_S signing queries and Q_H hash queries and outputs Q_S+1 signatures with probability δ, then there exist algorithms $\mathcal{B}$ and $\mathcal{C}$, both having essentially the same runtime as $\mathcal{A}$, where $\mathcal{B}$ requests Q_S preimage queries and wins the* $\mathsf{one\text{-}more\text{-}ISIS}_{q,n,m,\sigma,2\beta}$ *game with probability* $\mathsf{Adv}_{\mathsf{omisis}}$ *and $\mathcal{C}$ distinguishes the public key of the* PKE *scheme from uniform with advantage* $\mathsf{Adv}_{\mathsf{PKE}}$ *such that*

$$\mathsf{Adv}_{\mathsf{PKE}} + \mathsf{Adv}_{\mathsf{omisis}} \geq \delta - 2^{-\Omega(\lambda)} - (Q_S+1)(2^{-\Omega(\lambda)} + q^{-n}) - Q_H^2 q^{-n}.$$

Proof. We construct the proof using the following hybrids.

Hybrid_0: This is the genuine one more unforgeability experiment.
Hybrid_1 : In this hybrid, the challenger computes $\mathsf{PKE.pk}$ differently. It first runs $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and then programs $\mathsf{H}_{\mathsf{PKE}}() = \mathsf{PKE.pk}$. It stores $\mathsf{PKE.sk}$.
Hybrid_2 : In this hybrid, for every signature $\sigma_j = (\pi_j, \mathsf{ct}_j)$ output by the adversary (for $j \in [Q_S+1]$), the challenger uses $\mathsf{PKE.sk}$ to decrypt ct_j into a plaintext $(\mathbf{x}_j \| \mathbf{y}_j)$ (which can be $\perp$ in case decryption fails). It stores the $(\mathbf{x}_j \| \mathbf{y}_j)$'s.
Hybrid_3 : This hybrid differs from the previous hybrid in the way matrix $\mathbf{A}$ is computed. In this hybrid, the challenger first samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and then programs $\mathsf{H}_{\mathsf{A}}() = \mathbf{A}$.
Hybrid_4 : 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

In the following, we let $\mathsf{Adv}_i^{\mathsf{omuf}}$ represent the advantage of $\mathcal{A}$ in the one more unforgeability game in Hybrid_i. Then $\mathsf{Adv}_0^{\mathsf{omuf}} = \delta$.

1. The only difference between Hybrid_0 and Hybrid_1 is in the way $\mathsf{PKE.pk}$ is computed. Hence if $\mathcal{A}$ succeeds in the one more unforgeability game in Hybrid_1 with probability $\mathsf{Adv}_1^{\mathsf{omuf}}$, then there exists an algorithm $\mathcal{C}$ which runs in essentially the same time as $\mathcal{A}$ and distinguishes the public key of the PKE scheme from uniform with advantage at least $\mathsf{Adv}_0^{\mathsf{omuf}} - \mathsf{Adv}_1^{\mathsf{omuf}}$. Let us denote the advantage of $\mathcal{C}$ with $\mathsf{Adv}_{\mathsf{PKE}}$. Then $\mathsf{Adv}_1^{\mathsf{omuf}} \geq \mathsf{Adv}_0^{\mathsf{omuf}} - \mathsf{Adv}_{\mathsf{PKE}} = \delta - \mathsf{Adv}_{\mathsf{PKE}}$.
2. The differences between Hybrid_1 and Hybrid_2 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. Thus $\mathsf{Adv}_2^{\mathsf{omuf}} = \mathsf{Adv}_1^{\mathsf{omuf}} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}}$.
3. The only difference between Hybrid_2 and Hybrid_3 is that in the latter $\mathbf{A}$ is first chosen uniformly from $\mathbb{Z}_q^{n \times m}$ and then $\mathsf{H}_{\mathsf{A}}()$ is programmed to be $\mathbf{A}$. Hence, the two hybrids are identical in the adversary's view in the random oracle model. Thus $\mathsf{Adv}_3^{\mathsf{omuf}} = \mathsf{Adv}_2^{\mathsf{omuf}} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}}$.
4. The only difference between Hybrid_3 and Hybrid_4 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 Lemmas 2.10 and 2.11 imply that $(\mathbf{C}, \mathbf{A})$ is within statistical distance $2^{-\Omega(\lambda)}$ from $(\mathbf{C}, \mathbf{CR})$. Thus $\mathsf{Adv}_4^{\mathsf{omuf}} \geq \mathsf{Adv}_3^{\mathsf{omuf}} - 2^{-\Omega(\lambda)} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}} - 2^{-\Omega(\lambda)}$.

We conclude with the following claim.

Claim A.2 *Assume that the* NIZK *argument system is sound and* PKE *is perfectly correct. Then if there is an adversary $\mathcal{A}$ in the random oracle model that makes at most Q_H hash queries and Q_S signing queries and succeeds in generating Q_S+1 signatures in* Hybrid_4 *with probability ε, then there exists a* one-more-ISIS *adversary $\mathcal{B}$ with Q_S preimage queries with success probability at least $\varepsilon - (Q_S+1)(2^{-\Omega(\lambda)} + q^{-n}) - Q_H^2 q^{-n}$.*

Proof. The reduction $\mathcal{B}$ is as follows.

1. Upon being challenged by the one-more-ISIS challenger $\mathcal{C}$, with matrix $\mathbf{C}$, algorithm $\mathcal{B}$ does the following:
   - It uniformly samples a binary matrix $\mathbf{R}$ and sets $\mathbf{A} = \mathbf{CR}$. It programs $\mathsf{H_A}() = \mathbf{A}$.
   - It samples $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and sets $\mathsf{H_{PKE}}() = \mathsf{PKE.pk}$.
   - It invokes $\mathcal{A}$ with $\mathbf{C}$ as the 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 $\mathbf{Cy}' = \mathbf{t}'$. Algorithm $\mathcal{B}$ forwards $\mathbf{y}'$ to $\mathcal{A}$.
4. Eventually, adversary $\mathcal{A}$ outputs $Q_S + 1$ message-signature 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 (A.1) for all $j \in [Q_S + 1]$, then $\mathcal{B}$ outputs $\{(\mathbf{y}_j - \mathbf{Rx}_j, H(\mu_j))\}_{j \in [Q_S+1]}$. If any decryption fails or any of the above conditions is not satisfied, then $\mathcal{B}$ aborts.

First note that by the soundness of NIZK, the probability that a statement with a valid proof is false is bounded above by $2^{-\Omega(\lambda)}$. Now, since the ciphertext ct is part of the statement, we have by perfect correctness of PKE that it decrypts to the correct value. Hence, the overall probability that a decryption fails is $\leq (Q_S + 1) \cdot 2^{-\Omega(\lambda)}$.

Next, we claim that for each μ_j, adversary $\mathcal{A}$ must have issued a corresponding hash query to $\mathcal{B}$. This is because otherwise, there is only a q^{-n} probability that a fresh $H(\mu_j)$ is equal to $\mathbf{Cy}_j - \mathbf{Ax}_j$. Additionally, by the soundness of NIZK, it holds that for all $j \in [Q_S + 1]$:

$$\|\mathbf{x}_j\| \leq \beta/m \ \wedge \ \|\mathbf{y}_j\| \leq \beta \ \wedge \ \mathbf{Cy}_j - \mathbf{Ax}_j = H(\mu_j).$$

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 $j \in [Q_S + 1]$,

$$\mathbf{t}_j = \mathbf{Cy}_j - \mathbf{Ax}_j = \mathbf{Cy}_j - \mathbf{CRx}_j = \mathbf{C}(\mathbf{y}_j - \mathbf{Rx}_j).$$

Since $\mathbf{R}$ is a binary matrix, we have $\|\mathbf{y}_j - \mathbf{Rx}_j\| \leq 2\beta$ for all j. Finally, since all μ_j's are distinct as per Definition 2.2 and $\mathcal{B}$ answers each fresh hash query with a fresh syndrome query, the $\mathbf{t}_j$'s are distinct except with probability at most Q_H^2/q^n. Thus, the success probability of $\mathcal{B}$ is at least $\varepsilon - (Q_S+1)(2^{-\Omega(\lambda)} + q^{-n}) - Q_H^2 q^{-n}$.

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 $\mathcal{C}$. Hence $\mathcal{B}$ is a valid adversary in the one-more-ISIS game.

Theorem A.3. *Assume that* NIZK *is zero-knowledge. Then if there exists a signer $\mathcal{S}^*$ in the random oracle model that wins the full-fledged blindness game for the blind signature scheme in Figure 5 with advantage δ, then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$, both running in essentially the same time as $\mathcal{S}^*$, where $\mathcal{B}$ wins the* IND-CPA *security game for* PKE *with advantage* $\mathsf{Adv_{IND\text{-}CPA}}$ *and $\mathcal{C}$ distinguishes the public key of the* PKE *scheme from uniform with advantage* $\mathsf{Adv_{PKE}}$ *such that*

$$2\mathsf{Adv_{IND\text{-}CPA}} + \mathsf{Adv_{PKE}} \geq \delta - 2^{-\Omega(\lambda)}.$$

Proof. We argue blindness using the following hybrids.

Hybrid_0 : This is the genuine full-fledged blindness experiment.

Hybrid_1 : This hybrid differs from the previous one in the way $\mathsf{PKE.pk}$ is computed: the challenger now samples $(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$ and sets $\mathsf{H_{PKE}}() = \mathsf{PKE.pk}$.

Hybrid_2 : This hybrid differs from the previous one in the way the proofs π_0 and π_1 are computed: instead of genuinely computing the NIZKs, the challenger simulates them without using the witnesses.

Hybrid_3 : 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_4 : This hybrid differs from the previous hybrid in the way matrix $\mathbf{A}$ is computed. In this hybrid, the challenger first samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and then programs $\mathsf{H}_\mathsf{A}() = \mathbf{A}$.

Hybrid_5 : 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

In the following, we let $\mathsf{Adv}_i^{\mathsf{bl}}$ represent the advantage of $\mathcal{S}^*$ in the full-fledged blindness game in Hybrid_i. Then $\mathsf{Adv}_0^{\mathsf{bl}}$ is δ.

1. The only difference between Hybrid_0 and Hybrid_1 is in the way $\mathsf{PKE.pk}$ is computed. Hence if $\mathcal{S}^*$ succeeds in the full-fledged blindness game in Hybrid_1 with advantage $\mathsf{Adv}_1^{\mathsf{bl}}$, then we can design an algorithm $\mathcal{C}$ which runs in essentially the same time as $\mathcal{S}^*$ and distinguishes the public key of the PKE scheme from uniform with advantage at least $\mathsf{Adv}_0^{\mathsf{bl}} - \mathsf{Adv}_1^{\mathsf{bl}}$. Thus if the advantage of $\mathcal{C}$ is represented by $\mathsf{Adv}_{\mathsf{PKE}}$, we get $\mathsf{Adv}_1^{\mathsf{bl}} \geq \mathsf{Adv}_0^{\mathsf{bl}} - \mathsf{Adv}_{\mathsf{PKE}} = \delta - \mathsf{Adv}_{\mathsf{PKE}}$.
2. The only difference between Hybrid_1 and Hybrid_2 is in the way π_0 and π_1 are computed. The two hybrids are indistinguishable because of the zero-knowledge property of the NIZK. Hence $\mathsf{Adv}_2^{\mathsf{bl}} \geq \mathsf{Adv}_1^{\mathsf{bl}} - 2^{-\Omega(\lambda)} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}} - 2^{-\Omega(\lambda)}$.
3. The only difference between Hybrid_2 and Hybrid_3 is in the messages being encrypted by ct_0 and ct_1. The two hybrids are computationally indistinguishable because of the IND-CPA security of PKE. In particular, if advantage of $\mathcal{S}^*$ in the full-fledged blindness game in Hybrid_3 is $\mathsf{Adv}_3^{\mathsf{bl}}$, then there exists an adversary $\mathcal{B}$ against $\mathsf{IND\text{-}CPA}$ security of PKE with advantage $\mathsf{Adv}_{\mathsf{IND\text{-}CPA}}$ such that $2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} \geq \mathsf{Adv}_2^{\mathsf{bl}} - \mathsf{Adv}_3^{\mathsf{bl}} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}} - 2^{-\Omega(\lambda)} - \mathsf{Adv}_3^{\mathsf{bl}}$. (Here, we consider twice of $\mathsf{Adv}_{\mathsf{IND\text{-}CPA}}$ since both ct_0 and ct_1 are replaced with encryptions of $\mathbf{0}$ and hence the $\mathsf{IND\text{-}CPA}$ security of PKE is called twice.).
4. The only difference between Hybrid_3 and Hybrid_4 is in the computation of matrix $\mathbf{A}$: the challenger first samples $\mathbf{A}$ uniformly from $\mathbb{Z}_q^{n\times m}$ and then programs $\mathsf{H}_\mathsf{A}() = \mathbf{A}$. The two hybrids are therefore, identical in the adversary's view in the random oracle model. Hence $\mathsf{Adv}_4^{\mathsf{bl}} = \mathsf{Adv}_3^{\mathsf{bl}} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}} - 2^{-\Omega(\lambda)} - 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}}$.
5. The only difference between Hybrid_4 and Hybrid_5 is in the choice of the masking term for $H(\mu)$. Since the vectors $\mathbf{x}_0$ and $\mathbf{x}_1$ are only used in the computations of the vectors $\mathbf{t}_0$ and $\mathbf{t}_1$, we have by the leftover hash lemma (Lemmas 2.10 and 2.11), that $\mathbf{A}\mathbf{x}_0$ and $\mathbf{A}\mathbf{x}_1$ are statistically indistinguishable from uniform $\mathbf{u}_0$ and $\mathbf{u}_1$. Hence Hybrid_4 and Hybrid_5 are indistinguishable. More concretely, we have $\mathsf{Adv}_5^{\mathsf{bl}} \geq \mathsf{Adv}_4^{\mathsf{bl}} - 2^{-\Omega(\lambda)} \geq \delta - \mathsf{Adv}_{\mathsf{PKE}} - 2^{-\Omega(\lambda)} - 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} - 2^{-\Omega(\lambda)}$.

However, in Hybrid_5, the adversary $\mathcal{S}^*$ has zero advantage in guessing the bit b since it is information theoretically hidden. Hence $\delta - \mathsf{Adv}_{\mathsf{PKE}} - 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} - 2^{-\Omega(\lambda)} \leq 0$, which is equivalent to $\mathsf{Adv}_{\mathsf{PKE}} + 2\mathsf{Adv}_{\mathsf{IND\text{-}CPA}} \geq \delta - 2^{-\Omega(\lambda)}$.