Batched & Non-interactive Blind Signatures from Lattices*

Foteini Baldimtsi†
George Mason University
& Mysten Labs
foteini@gmu.edu

Rishab Goyal‡
University of Wisconsin–Madison
rishab@cs.wisc.edu

Aayush Yadav§
George Mason University
ayadav5@gmu.edu

Abstract

Non-interactive blind signatures (NIBS; Eurocrypt '23) allow a signer to asynchronously generate presignatures for a recipient, ensuring that only the intended recipient can extract a "blinded" signature for a random message.

We introduce a new generalization called non-interactive *batched* blind signatures (NIBBS). Our goal is to reduce the computation and communication costs for signers and receivers, by batching multiple blind signature queries. More precisely, we define the property of 'succinct communication' which requires that the communication cost from signer to receiver be *independent* of the batch size. NIBBS is very suitable for large-scale deployments requiring only minimal signer-side effort.

We design a NIBBS scheme and prove its security based on the hardness of lattice assumptions (in the random oracle model). When instantiated with the low-depth PRF candidate "Crypto Dark Matter" (TCC '18) and the succinct lattice-based proof system for rank-1 constraint systems (Crypto '23), our final signature size is 308 KB with < 1 KB communication.

1 Introduction

Blind signatures are a special class of digital signatures that enable a receiver to obtain a signature on a message without revealing the message to the signer. They were introduced by Chaum [Cha83] as a building block for private e-cash. The idea was that a bank, acting as the signer, signs a coin's serial number and some other useful information without seeing it, so that when the coin is later spent, the bank cannot link it to any specific user. Since then, blind

*This is the full version of the article of the same title published in the proceedings of IACR PKC '26.

†Support for this research was provided through NSF award 2143287.

‡Support for this research was provided by a gift from the Stellar Development Foundation.

§Work done while visiting UW–Madison. Supported by NSF award 2143287.

signatures have emerged as an attractive tool across a variety of applications. These include e-voting [CGT06], anonymous credentials [PZ13, BL13, FHS15, DGS+18], direct anonymous attestation [BCC04], and cryptocurrency mixers [HBG16, HAB+17].

Blind signatures are typically formalized as a two-party protocol between a *signer* (holding a secret key sk) and a *receiver* (holding a message μ and signer's verification key vk). In blind signatures, signing is an *interactive* process—the recipient sends a "blinded" message to the signer, and the signer responds with a signature on the blinded message. The recipient then locally "unblinds" the signature and outputs the final message-signature pair (μ, σ) which can be verified using vk. The two central security properties are *blindness* (the signer learns nothing about μ and cannot link the transcripts of protocol executions to the signatures that it created) and *one-more unforgeability* (if the receiver is granted Q signatures, it cannot produce $Q+1$ or more distinct valid signatures) [JLO97].

Non-interactive blind signatures. Blind signatures are inherently interactive protocols, making *two-move* protocols *round-optimal*. However, the two-move barrier holds only if the message to be signed is selected by the receiver. In many applications, this is not the case! For instance, in e-cash, coins are simply signatures on random coin IDs. Thus, a natural question is whether we can design *non-interactive* blind signatures, when the messages are random strings.

This was studied in the groundbreaking work of Hanzlik [Han23]. Clearly, a non-interactive blind signature (NIBS) will lead to *minimal round complexity* (just one-move, from signer to receiver). Hanzlik pointed out that NIBS could readily replace standard blind signatures in any application where the message choice was unimportant and could be random/unstructured. Interestingly, there are many such applications beyond e-cash. For instance, NIBS can be used for anonymous token systems (à la Privacy Pass [DGS+18]), lottery systems [Han23] and cryptocurrency airdrops [BCGY24], where coins are distributed by creating and publishing a pre-signature, enabling users to privately obtain the final signatures (coins). Among these, anonymous token systems stand out as a particularly compelling application, given the growing industry interest[1] and the ongoing standardization efforts through IETF [IET25b].

Batching non-interactive blind signatures. In many applications, users require *multiple tokens at once* rather than a single token each time they contact the issuer. For instance, this happens in Privacy Pass [DGS+18], which was originally proposed for anonymous authentication in content delivery networks (CDNs). In light of many such emerging applications, the concept of batching in (interactive) blind signatures was recently introduced by Hanzlik, Loss and Wagner [HLW23]. The importance of batched token issuance is equally reflected in ongoing standardization efforts, such as a recent IETF draft [IET25a], which proposes batched issuance techniques for privacy-preserving tokens.

Somewhat unfortunately, the [HLW23] construction only offers amortized cost benefits in the interactive setting. This does not come as a surprise given that, in regular blind signatures, the messages are selected by the receiver, and for each message the signer has to compute a separate response. Thus, the communication complexity (both, Receiver $\to$ Signer and Signer $\to$ Receiver) scales linearly with the batch size.

Our observation is that *batching* makes significantly more sense in the non-interactive setting. In NIBS, a receiver/client could simply specify the number of tokens/signatures it wants (say N),

[1] Major technology companies have been actively developing such systems, including Google's Private State Tokens [Goo25] and Cloudflare's Privacy Pass [MRFH25].

and the signer could (potentially) issue a single pre-signature that can be "un-blinded" to obtain N blind signatures. Since each blind signature is supposed to be associated to a random message, we could hope to keep the communication cost to be *sub-linear* (and ideally constant) with respect to the batch size. As such, NIBS already eliminates the need for back-and-forth communication between signer and receiver, and we believe a batched variant would significantly bring down computation and communication costs, enabling large-scale deployments with minimal signer-side effort.

The major question that inspires this work is thus:

Q. Can we design a practical batched NIBS scheme with provable security, while keeping communication cost independent of batch size?

Our results. In this work, we introduce the concept of *batching* in NIBS and define the notion of non-interactive batched blind signatures (NIBBS). We provide a practical and post-quantum NIBBS construction, based on hardness of lattice problems.

Taking the batch size to be 1, our NIBBS scheme also gives the first concretely efficient post-quantum secure NIBS scheme. Prior to this work, the known post-quantum constructions either required a relaxed notion of blindness [BCGY24] or gave up concrete efficiency [BCGY24, ZCH25]. In other words, we did not have a candidate construction for NIBS that was *practical and post-quantum*. We summarize our contributions below:

❶ **Formalizing and constructing non-interactive batched blind signatures.** We formally introduce the notion of batching in NIBS. We then provide a NIBBS construction and prove it secure under the hardness of the randomized one-more ISIS assumption [BCGY24] (and other standard lattice assumptions [Ajt96, Reg05]) in the random oracle model [BR94]. Notably, the communication cost of our scheme is independent of the batch size, and at just under 1 KB, is nearly optimal. Current conservative estimates put the size of a single blind signature to be around 308 KB, depending on the parameter choices. We also present an alternate construction from the ISIS_f assumption [BLNS23b] that bypasses some of the technical shortcomings of the previous approach.

❷ **Practical-ish post-quantum NIBS.** Using the same instantiation, we also obtain the first concretely efficient post-quantum (non-batched) NIBS scheme that is provably secure under the same set of assumptions. As with the batched scheme, the overall communication is minimal (< 1 KB), and our estimate puts the size of the final blind signature to be between slightly smaller at 306 KB.

1.1 Related work

Since the introduction of blind signatures, numerous schemes have been designed based on various number-theoretic assumptions. Some constructions are based in the plain model [GRS+11, BFPV13, GG14, Gha17], while many practical schemes rely on idealized models for their security proofs [PS96, PS00, Abe01, Bol03, BL13, FHS15, HKL19, KLR21, KLX22, CAHL+22, TZ22, CATZ24]. We also have generic constructions due to Juels et al. [JLO97] and Fischlin [Fis06], who developed the first generic blind signature scheme achieving optimal round complexity.

In recent years, several round-optimal schemes from lattices have been proposed [LNP22a, dPK22, AKSY22, BLNS23a]. Lyubashevsky et al. [LNP22a] introduced a scheme using one-time

signatures under Module-SIS and Module-LWE assumptions, but it supports only a bounded number of signatures per key, with each signature around 150 KB. Del Pino and Katsumata [dPK22] adapted Fischlin's paradigm for unbounded signatures, reducing the size to 102 KB. Agrawal et al. [AKSY22] used lattice-based NIZKs to shrink signature size to 45 KB and transcript size to 1 KB relying on the one-more ISIS assumption [AKSY22]. Beullens et al. [BLNS23a] improved this by shifting costly computation to the receiver, reducing the signature size to 22 KB but increasing the first message to several hundred kilobytes. Most recently, Jeudy and Sanders [JS24] showed that it was possible to reuse randomness from commitments to mask pre-signature, achieving a 41 KB signature size, but with faster signing than previous works.

Scheme	Post-quantum secure	#Moves	Communication $N = 4$	$N = 16$	$N = 256$	$\lvert\sigma\rvert$
[HLW23]	✗	3	67.92–175.88	206.72–588.32	2982.4–8837.12	9.41–13.98
This work (NIBBS, §5)	✓	1	——— 0.74 ———			308.78

Table 1: Communication and signature sizes of practical batched blind signature schemes (in kilobytes). Communication is given for batches of N messages. Horizontal line through a row indicates that the number is common across all batch sizes.

Non-interactivity. The notion of non-interactive blind signatures (NIBS) was introduced by Hanzlik [Han23], who proposed a generic construction using verifiable random functions, digital signatures, and dual-mode non-interactive witness indistinguishable proofs, proving security in the random oracle model (RO). Hanzlik also developed a practical NIBS scheme based on signatures on equivalence classes [HS14, FHS19], with security proven in the generic group model (GGM). All schemes in [Han23] were proved to satisfy weak blindness. Building on this, Baldimtsi et al. [BCGY24] identified limitations in the original definitional framework [Han23], proposing an improved framework with stronger security definitions. They presented two new constructions: a generic paradigm using circuit-private leveled homomorphic encryption (LHE) and a simpler approach based on general-purpose non-interactive zero-knowledge (NIZK) proofs. Additionally, they introduced a practical lattice-based NIBS scheme under a new randomized one-more-ISIS (rOM-ISIS) assumption, though it only satisfied the weaker security properties from the original framework.

Recently, Zhang, et al. [ZCH25] gave a construction for lattice-based NIBS in the random oracle model by partially instantiating the generic VRF construction of [Han23] and proved blindness under the weaker definitions. This construction was proven insecure by a concurrent work [GKK+25], who also gave an updated construction which is claimed to to be secure under the rOM-ISIS assumption. We do not include this work in our comparisons as it does not give concrete estimates, however we note that the specific VRF instantiation that the authors propose [EKS+21] yields a scheme with highly limited reusability as [EKS+21] only supports up to 5 VRF evaluations under the same key. This can be circumvented with the lattice-based VRF of Esgin, et al. [ESLR23] at the cost of two additional proofs of hash computations, which leads us to believe that our scheme is concretely more efficient.

Finally, Hanzlik, et al. [HPZ25] recently proposed a generic construction using garbled circuits, which can be instantiated to support pre-signature issuance on standard RSA public keys.

Scheme	Assumption	Communication R → S	Communication S → R	$\lvert\sigma\rvert$
[AKSY22]	OM-ISIS	0.96	0.56	45
[BLNS23a]	MLWE, MSIS	∼ 50	∼ 60	22
[JS24]	MLWE, MSIS	8.73	50.89	41.12
This work (NIBS)	rOM-ISIS	—	0.68	306.18

Table 2: Comparison of state-of-the-art practical lattice-based blind signatures, all sizes in kilobytes. R → S indicates the total communication from the recipient to the signer (similarly S → R. [BCGY24] is omitted from the list as it does not achieve full blindness.

Batching. The concept of batching in blind signatures was introduced by Hanzlik, et al. [HLW23]. Their main result was a concurrently secure round-optimal blind signature based on cut-and-choose. In addition, they introduced a formal model for batched blind signatures, and proposed a batching technique for their scheme in order to achieve reduced (amortized) communication complexity.

We give a comparison of our NIBBS scheme with other batched blind signatures in Table 1 and of our NIBS scheme with other state-of-the-art lattice-based blind signatures in Table 2.

2 Technical Overview

We introduce a new generalization for NIBS, that we call non-interactive *batched* blind signatures (NIBBS). Briefly, the goal in NIBBS is to provide a new capability by allowing a receiver to derive an entire batch of (blind) message-signature pairs from a single pre-signature. As suggested earlier, this drastically reduces the overall communication size between signer and receiver for applications where a receiver needs more than one blind signature.

Formalizing NIBBS. For simplicity, consider a globally fixed batch size N. Syntactically, a NIBBS scheme (similar to NIBS) consists of the following algorithms: the Setup algorithm generates the system's global public parameters pp, which serves as a common reference string (CRS); two key generation algorithms $\mathsf{KeyGen}_S \to (\mathsf{sk}, \mathsf{vk})$ for the signer and $\mathsf{KeyGen}_R \to (\mathsf{sk}_R, \mathsf{pk}_R)$ for the receiver; using the secret key sk, the signer runs the randomized Issue algorithm for any receiver's public key pk_R to compute a pre-signature psig and a nonce nonce, which represents (a portion of) the signer's randomness; finally, the receiver's Obtain algorithm outputs N message-signature pairs $[\![\mu, \sigma]\!] := \{(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_N, \sigma_N)\}$.

Baldimtsi, et al. [BCGY24] introduced the notion of reusability for NIBS. Informally, it requires that a signer be able to issue multiple distinct pre-signatures (leading to multiple distinct message-signature pairs) for the *same* receiver public key pk_R. Clearly, without reusability a NIBS would be somewhat ineffectual. As with standard NIBS, we define reusability in the batched setting to guarantee that a signer is able to issue multiple pre-signatures to the same receiver public key pk_R, and further that the receiver can obtain N distinct final message-signature pairs for each pre-signature.

In terms of security, NIBBS should satisfy the properties of *one-more unforgeability* and *blindness*. One-more unforgeability in the NIBBS context is a natural extension of the same property for

NIBS. Namely, the adversary, after seeing Q pre-signatures for receiver public keys of its choice, must produce $(NQ+1)$ valid signatures.

Blindness ensures that a final signature cannot be linked to its corresponding pre-signature. However, since the receiver now obtains an entire batch of signatures from a single pre-signature, we must ensure that our definition not only captures the unlinkability between signatures from different batches, but also between signatures *within* the same batch. To that end, we define two security notions—*inter-batch blindness* and *intra-batch blindness*. For the inter-batch blindness experiment, the changes from the non-batched setting are minimal. Specifically, the adversary is given access to the Obtain oracle with respect to two different receiver secret keys, sk_{R_0} and sk_{R_1}, and outputs two pairs of $(\mathsf{psig}_b, \mathsf{nonce}_b)_{b\in\{0,1\}}$. The key difference in the batched setting is that the challenger now runs the Obtain algorithm on each $(\mathsf{psig}_b, \mathsf{nonce}_b)$, and provides the adversary with two batches of message-signature pairs, $[\![\mu, \sigma]\!]_{\hat{b}}$ and $[\![\mu, \sigma]\!]_{1-\hat{b}}$ for a random bit $\hat{b}$. The adversary's goal remains to correctly map each batch to their corresponding pre-signature, i.e., to guess the bit $\hat{b}$.

For intra-batch blindness, the adversary is given access to the Obtain oracle with respect to a single receiver secret key, sk_R, and outputs two pairs of $(\mathsf{psig}_b, \mathsf{nonce}_b)_{b\in\{0,1\}}$ along with a "permutation" map, φ. The challenger runs the Obtain algorithm on each $(\mathsf{psig}_b, \mathsf{nonce}_b)$ and provides the adversary with the resulting $2N$ message-signature pairs, either in the original order if the random bit $\hat{b} = 0$, or permuted according to φ otherwise. Once again, the goal of the adversary is to correctly guess $\hat{b}$. Importantly, this ensures that an adversary cannot link final signatures to their respective batches, thus giving us the aforementioned "intra"-batch blindness property.

Constructing NIBBS. The starting point for our design is the generic template for NIBS given in [BCGY24]. The core idea is as follows: the receiver's public key pk_R is set as a commitment to a pseudorandom function (PRF) key K, while sk_R is set to be K along with the opening randomness for the commitment. To issue a pre-signature, the signer samples a random string r and signs $\mathsf{pk}_R \| r$, sending both the (pre)signature as well as the random nonce r. At the other end, the receiver computes the message μ as a PRF evaluation of r under key K, and the corresponding signature as a proof of knowledge of a pre-signature on $\mathsf{pk}_R \| r$ such that pk_R is a commitment to K and μ is a PRF evaluation of r under K.

The binding property of the commitment scheme ensures that, with high probability, the receiver computes a single signature for a given pre-signature and nonce pair, then by NIZK soundness[2] and unforgeability of the underlying signature scheme, we can prove one-more unforgeability of NIBS. Blindness for this design follows from simpler reductions to the (multi-theorem) zero-knowledge property of NIZKs and commitment hiding security.

As it turns out, this NIBS template can be quite naturally generalized to support *batching*. In particular, instead of evaluating the PRF on nonce r, the receiver evaluates it on $r\|\mathsf{i}$ for each $\mathsf{i} \in [N]$, thus obtaining N distinct messages, and their corresponding signatures. In order to guarantee unforgeability, the statement for the ith batch must additionally require that $\mathsf{i} \leqslant N$, with i as part of the witness.

Lattice-based instantiation. We now explain how to instantiate the generic approach essentially starting from the template of [AKSY22, BCGY24]. Let the vector $\mathbf{k}$ of small norm be the receiver's secret PRF key. Then, for a random vector $\mathbf{u}$, and public matrix $\mathbf{C}$, the receiver's public key pk_R is

[2] To be more precise, this requires the stronger knowledge soundness, which can be generically achieved in the CRS model by relying on public-key encryption.

given by $\mathbf{t} := \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$.

Given $\mathbf{t}$, the signer with public verification key matrices $\mathbf{A}$ and $\mathbf{B}$ can now create a pre-signature by (pre-)signing $\mathbf{t}$ along with a short random nonce vector $\mathbf{r}$ by using trapdoor lattice sampling [GPV08] (see also [AKSY22, BLNS23a, BCGY24]) to sample a short preimage $\mathbf{s}$ with respect to $\mathbf{A}$ such that $\mathbf{A} \cdot \mathbf{s} + \mathbf{B} \cdot \mathbf{r} = \mathbf{t}$ and send, both, pre-signature $\mathbf{s}$ and nonce $\mathbf{r}$ to the receiver.

On receiving the pre-signature and nonce pair, the receiver can compute N random messages as the PRF evaluation $\mu_i := \mathsf{F}_{\mathbf{k}}\left(\mathbf{r} \| \vec{i}\right)$ (where $\vec{i}$ is the binary representation of $i \in [N]$), and the corresponding signatures as the NIZK proof π_i stating that, given $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and μ_i, there exist vectors $\mathbf{k}, \mathbf{u}, \mathbf{s}$ and $\mathbf{r}$, and integer i such that $\mathbf{A} \cdot \mathbf{s} + \mathbf{B} \cdot \mathbf{r} = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\mu_i = \mathsf{F}_{\mathbf{k}}\left(\mathbf{r} \| \vec{i}\right)$, $i \in [N]$ and $\|\mathbf{k}\|, \|(\mathbf{s}, \mathbf{r})\|$ are short.

A technical limitation for such Fischlin-style approaches is that the NIZK must be straight-line extractable. Fortunately, we can utilize the folklore "encryption-to-the-sky" trick and have the receiver encrypt its secrets using public key encryption. In particular, this allows a challenger, holding the secret decryption key $\mathsf{sk}_{\mathsf{PKE}}$, to extract all the secrets in a straight-line manner. Let ψ_i be the encryption of the witness to the relation under public key $\mathsf{pk}_{\mathsf{PKE}}$, sampled during setup, and let ρ_i denote the encryption randomness. Then, each NIZK must also prove correct computation of ψ_i where ψ_i and $\mathsf{pk}_{\mathsf{PKE}}$ are part of the instance, and ρ_i of the witness. Importantly, the final signature must now also include ψ_i.

Estimating efficiency. Efforts to construct practically efficient (non-)interactive blind signature schemes from hard lattices assumptions [AKSY22, BLNS23a, BCGY24] have largely relied on the use of the NIZK proof system of Lyubashevsky, et al. [LNP22b] which produces very short proofs for affine relations. It is therefore crucial in these works that the entire NIZK relation is also composed of affine equations (and norm bounds). Thus, in order to build practical lattice-based NIBBS, we would need to be able to specify the lattice-instantiation explicitly in linear terms. However, for the security of the protocol we need to be able to prove that the final (public) message μ_i was computed as a PRF evaluation of $(\mathbf{r} \| \vec{i})$ with key $\mathbf{k}$ where, in particular, $\mathbf{r}$ and $\mathbf{k}$ are hidden as part of the witness of the NIZK (otherwise, blindness can be trivially broken).

This underpins the core technical hurdle in instantiating the generic template for NIBBS via affine relations in order to apply the [LNP22b] framework. Namely, that all popular approaches for designing secure PRFs from lattice-based assumptions require very high degree computations in the inputs and PRF keys [BPR12, BLMR13, BP14]. This technical tension suggests that any natural attempt in instantiating the generic template using practical lattice-based NIZKs will ultimately fail[3].

We remark that for many existing lattice-based PRFs [BPR12, BLMR13, BP14], their evaluation can be segmented into $\sim \lambda$ incremental checks of degree-1 (or 2) such that by linking all $O(\lambda)$ checks together, it can be proven using the [LNP22b] framework that the PRF was correctly computed (this approach was used in [ADDS21, AG24]). Unfortunately, the family of lattice-based PRFs involve rounding down the final computation (modulo q), to a lower modulus p such that q/p is superpolynomial in the security parameter; an artifact of the reduction from the learning with rounding (LWR) assumption, on which they rely, to the learning with errors (LWE) assump-

[3] This is in part why Baldimtsi et al. [BCGY24] could only design lattice-based NIBS with *very weak security* [Han23]. They simply omitted the PRF computation entirely, and replaced it with a random linear function. They proved that as long as blindness is only desired against at most two signatures, then their NIBS are secure. Moreover, they suggested that one could easily break blindness of their lattice-based NIBS protocol if an attacker learns as few as *three* blind signatures.

tion. Consequently, the modulus for the NIZK proof system, which itself has to be greater than q for soundness, is rather large in practice.

Given, these challenges inherent in instantiating the protocol, we make two crucial design choices intended to make our final approach practical. Firstly, we instantiate our scheme with the Crypto Dark Matter PRF candidate [BIP+18] which can be computed by depth-3 arithmetic circuit. Secondly, we use the lattice-based general purpose SNARK LaBRADOR [BS23] which provides compact proofs for rank-1 constraint systems (R1CS) to avoid any dependence on [LNP22b][4].

In Section 5.2, we provide concrete instantiation of parameters for our NIBBS construction. The estimated sizes of the pre-signature and the final blind signatures are provided in Tables 1 and 2. Our estimates are not based on aggressive calculations, nor assume any non-standard software optimization tricks. We simply provide our estimates as a proof-of-concept to show practical feasibility of NIBBS, and we believe that our framework will serve as an important stepping stone for future research in lattice-based (non-interactive) batched blind signatures.

Remark 2.1 (Fully dynamic batching)**.** While our above exposition considers a fixed batch size N, our approach is much more general. In particular, we can design NIBBS with truly dynamic batch sizes. For instance, consider that each receiver–when it contacts a signer–provides a (user-specific) batch size N to the signer. Now to handle such adaptive batch sizes, we make a small adjustment to the pre-signature computation. Instead of creating a (standard) digital signature for just the receiver's public key $\mathsf{pk}_R := \mathbf{t}$, it can create the signature for $\mathsf{pk}_R \| N$. Now, during the final blind signature generation, the receiver treats N as part of the NP witness. This readily gives us NIBBS with fully dynamic batch size. In the main body, we consider static batch sizes for ease of exposition, however our ideas can be easily generalized as explained here.

3 Preliminaries

Notation. We use λ to denote the security parameter. We use $\leftarrow\!\!\$$ to denote the output of a randomized algorithm and $\leftarrow$ to denote output of a deterministic algorithm. We denote the set of all positive integers up to n as $[n] := \{1, \ldots, n\}$ and the set of all non-negative integers up to n as $[0, n] := \{0\} \cup [n]$.

Following common convention, we use lowercase bold-face letters to denote vectors and uppercase bold-face letters for matrices. For a vector $\mathbf{x}$, x_i denotes its ith element. For a vector $\mathbf{x}$, we write its ℓ_2 norm as $\|\mathbf{x}\|_2$, often dropping the subscript and writing it simply as $\|\mathbf{x}\|$. We write the ℓ_∞ norm of $\mathbf{x}$ as $\|\mathbf{x}\|_\infty$.

We write polynomials $u(X)$ in indeterminate X simply as u when it is clear from context. In all that follows, $\mathbb{Z}$ is the set of integers, and $\mathbb{Z}_q$ denotes the set of integers modulo q. For a power of two d, we denote by $\mathcal{R}$ the ring $\mathbb{Z}[X]/(X^d + 1)$. We write the ring $\mathbb{Z}_q[X]/(X^d + 1) \cong \mathcal{R}/q\mathcal{R}$ as $\mathcal{R}_{q,d}$. When the degree is omitted from the subscript, it is assumed to be the fixed protocol parameter d.

Decomposition gadget. We will utilize a suitable "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times n \cdot \ell}$, such that there exists a deterministic inverse function $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \mathbb{Z}^{n \cdot \ell}$, with a "short" image and $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$ for any $\mathbf{x} \in \mathbb{Z}_q^n$. A typical example is $\mathbf{G} = \mathbf{I}_n \otimes \begin{bmatrix} 2^0 & 2^1 & \cdots & 2^{\ell-1} \end{bmatrix}$ with $\mathbf{G}^{-1}(\mathbf{x})$ defined such that each

[4] LaBRADOR can be made zero-knowledge with a small overhead (cf. [BS23]).

of its ℓ entries has $\{0,1\}$ coefficients. More simply, $\mathbf{G}^{-1}$ may be viewed as the (entry-wise) binary decomposition function.

Vectorization. The vectorization operation $\mathsf{Vec} : \mathbb{Z}^{n\times m} \to \mathbb{Z}^{nm}$ is a linear algebraic transformation which maps an $n \times m$ matrix to a unique vector of dimension nm. Let $\mathbf{e}_i$ be the ith canonical basis vector for the m-dimensional space, then for a row-major vectorization, it holds that [5] $\mathsf{Vec}(\mathbf{X}) = \sum_{i=1}^{m} (\mathbf{e}_i \otimes \mathbf{I}_n)\mathbf{X}\mathbf{e}_i$.

3.1 Randomness extraction

The min-entropy of a random variable X is defined as $\mathbf{H}_\infty(X) := -\log_2(\max_x \Pr[X = x])$. Let $\Delta(X, Y)$ denote the statistical distance between two random variables X and Y. Below we state the Leftover Hash Lemma (LHL) from [HILL99, DRS04, DRS04].

Lemma 3.1 (Leftover hash lemma)**.** *Let $\mathcal{H} = \{h : X \to Y\}$ be a universal hash family, then for any random variable W taking values in X, the following holds*

$$\Delta((h, h(W)), (h, U_Y)) \leqslant \frac{1}{2}\sqrt{2^{-\mathbf{H}_\infty(W)} \cdot |Y|},$$

where U_Y denotes uniform distribution over Y.

In our security proofs, we make use of a corollary following from the previous lemma.

Corollary 3.2. *Let $\ell > m \cdot n \log q + \omega(\log n)$, q a prime, and m, n are positive integers. Let $\mathbf{R}$ be an $k \times m$ matrix chosen as per distribution $\mathcal{X}$, where $k = k(n)$ is polynomial in n and $\mathbf{H}_\infty(\mathcal{X}) = \ell$. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times k}$ and $\mathbb{Z}_q^{n\times m}$, respectively. Then the statistical distance between the following distributions is negligible in n.*

$$\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R})\} \approx_s \{(\mathbf{A}, \mathbf{B})\}$$

3.2 Lattice preliminaries

Given positive integers n, m, q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we let $\Lambda_q^{\perp}(\mathbf{A})$ denote the m-dimensional lattice

$$\{\mathbf{x} \in \mathcal{R}^m \ : \ \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod q\} \ .$$

For $\mathbf{t} \in \mathbb{Z}_q^n$, we let $\Lambda_q^{\mathbf{t}}(\mathbf{A})$ denote the coset $\{\mathbf{x} \in \mathcal{R}^m \ : \ \mathbf{A} \cdot \mathbf{x} = \mathbf{t} \pmod q\}$.

Discrete Gaussians. Let $\mathfrak{s}$ be any positive real number. The discrete Gaussian distribution $\mathcal{D}_{\mathcal{R}^m,\mathfrak{s}}$ on $\mathcal{R}^m$ with parameter $\mathfrak{s}$ is defined by

$$\rho_{\mathcal{R}^m,\mathfrak{s}}(\mathbf{x}) = \frac{\exp(-\|\mathbf{x}\|^2/2\mathfrak{s}^2)}{\sum_{\mathbf{z}\in\mathcal{R}^m} \exp(-\|\mathbf{z}\|^2/2\mathfrak{s}^2)} \ .$$

The following lemma shows that if the parameter $\mathfrak{s}$ of a discrete Gaussian distribution is small, then any vector drawn from this distribution will be short (with high probability).

[5] Moreover, for $\mathbf{x} \in \mathbb{Z}^{nm}$ the inverse vectorization is given by $\mathsf{Vec}^{-1}(\mathbf{x}) := \left(\mathsf{Vec}(\mathbf{I}_m)^\top \otimes \mathbf{I}_n\right) \cdot (\mathbf{I}_m \otimes \mathbf{x})$.

Lemma 3.3 ([MR04, Lemma 4.4]). *Let m, n, q be positive integers with $m > n$, $q \geqslant 2$. Then for $\mathfrak{s} = \tilde{\Omega}(n)$ and any m-dimensional lattice $\mathcal{L} = \Lambda_q^{\perp}(\mathbf{A})$,*

$$\Pr\left[\|\mathbf{x}\| > \mathfrak{s}\sqrt{m} \ : \ \mathbf{x} \leftarrow\$ \mathcal{D}_{\mathcal{L},\mathfrak{s}}\right] \leqslant \mathsf{negl}(n) \ .$$

Lattice trapdoors. "Lattices with trapdoors" are statistically indistinguishable from randomly chosen lattices, but have certain trapdoors that allow efficient solutions to hard lattice problems [Ajt96, GPV08]. More formally, for lattice parameters n, m, q with $m \geqslant O(n \log q)$, a trapdoor lattice sampler consists of algorithms TrapGen and SamplePre with the following syntax[6]:

- $\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T_A})$. The lattice generation algorithm is a randomized algorithm that takes as input the matrix dimensions n, m, modulus q, and outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T_A}$.
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{t}, \mathfrak{s}) \to \mathbf{s}$. The presampling algorithm takes as input a matrix $\mathbf{A}$, trapdoor $\mathbf{T_A}$, a vector $\mathbf{t} \in \mathbb{Z}_q^n$ and a parameter $\mathfrak{s} \in \mathbb{R}_{>0}$ (which determines the length of the output vectors). It outputs a vector $\mathbf{s} \in \mathbb{Z}_q^m$ such that $\mathbf{A} \cdot \mathbf{s} = \mathbf{t}$ and $\|\mathbf{s}\| \leqslant \mathfrak{s}\sqrt{m}$.

Furthermore, these algorithms must satisfy the following properties:

i. **Well-distributedness of matrix.** The following distributions are statistically indistinguishable:
$$\{\mathbf{A} \ : \ (\mathbf{A}, \mathbf{T_A}) \leftarrow\$ \mathsf{TrapGen}(1^n, 1^m, q)\} \approx_s \left\{\mathbf{A} \ : \ \mathbf{A} \leftarrow\$ \mathbb{Z}_q^{n \times m}\right\} \ .$$

ii. **Preimage sampling.** For all $(\mathbf{A}, \mathbf{T_A}) \leftarrow\$ \mathsf{TrapGen}(1^n, 1^m, q)$, if $\mathfrak{s} = \omega\left(\sqrt{n \log q \cdot \log m}\right)$, then the following distributions are statistically indistinguishable:
$$\left\{\mathbf{s} : \mathbf{t} \leftarrow\$ \mathbb{Z}_q^n, \mathbf{s} \leftarrow\$ \mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{t}, \mathfrak{s})\right\} \approx_s \mathcal{D}_{\mathcal{R}^m, \mathfrak{s}} \ .$$

These properties are satisfied by the gadget-based trapdoor lattice sampler of [MP12] for parameters m such that $m = \Omega(n \log q)$. This was specifically shown for the case of module lattices in [BEP+21, Lemma 4] and [BEP+21, Theorem 4].

3.3 Hardness assumptions

Definition 3.4 (rOM-ISIS [BCGY24]). Let $q, n, m, \mathfrak{s}, \mathfrak{B}$ be functions of security parameter λ. Consider the following experiment:

i. The challenger uniformly samples two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$, and sends $\mathbf{A}, \mathbf{B}$ to adversary $\mathcal{A}$.

ii. $\mathcal{A}$ adaptively makes queries of the following types to the challenger, in any order.

   - **Syndrome queries.** $\mathcal{A}$ requests for a challenge vector, to which the challenger replies with a uniformly sampled vector $\mathbf{t} \leftarrow\$ \mathbb{Z}_q^n$. We denote the set of received vectors by $\mathcal{S}$.

[6] In certain places, we may choose to omit the trapdoor generation step and instead opt for the more succinct notation $\mathbf{s} \leftarrow\$ \mathbf{A}_{\mathfrak{s}}^{-1}(\mathbf{t})$ instead.

- **Preimage queries.** $\mathcal{A}$ queries a vector $\hat{\mathbf{t}} \in \mathbb{Z}_q^n$, to which the challenger replies with a short vector $\hat{\mathbf{s}} \in \mathbb{Z}_q^m$ and a ± 1 vector $\hat{\mathbf{r}} \in \mathbb{Z}_2^m$ such that $\mathbf{A} \cdot \hat{\mathbf{s}} + \mathbf{B} \cdot \hat{\mathbf{r}} = \hat{\mathbf{t}}$ and $\|\hat{\mathbf{s}}\| \leqslant \mathfrak{s}\sqrt{m}$. Let Q denote the total number of preimage queries.

iii. Finally, $\mathcal{A}$ outputs $Q+1$ tuples of the form $\left\{(\mathbf{s}_j, \mathbf{r}_j, \mathbf{t}_j)\right\}_{j \in [Q+1]}$. $\mathcal{A}$ wins if

$$\forall j \in [Q+1], \;\; \mathbf{A} \cdot \mathbf{s}_j + \mathbf{B} \cdot \mathbf{r}_j = \mathbf{t}_j, \; \left\|\mathbf{s}_j\right\| \leqslant \mathfrak{B}, \; \mathbf{r}_j \in \mathbb{Z}_2^m \text{ and } \mathbf{t}_j \in \mathcal{S} \;\; .$$

The $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}}$ assumption states that for every PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins is $\mathsf{negl}(\lambda)$.

3.4 Cryptographic building blocks

We recall the standard cryptographic building blocks that will be essential to this work.

3.4.1 Pseudorandom functions

A pseudorandom function (PRF) $\mathsf{F} : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a keyed function[7], that takes a λ-bit key as input, and on input $x \in \{0,1\}^\lambda$, it outputs a value $y = \mathsf{F}_K(x)$. *(Throughout the paper, we use $\mathsf{F}_K(x)$ as a shorthand for PRF evaluation.)*

Definition 3.5 (Pseudorandomness)**.** *A PRF F is said to be secure if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all λ, the following holds:*

$$\Pr\left[b = \mathcal{A}^{\mathcal{O}_{K,b}(\cdot)}(1^\lambda) \;:\; K \leftarrow\!\$\, \{0,1\}^\lambda, \; b \leftarrow\!\$\, \{0,1\}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda),$$

where oracle $\mathcal{O}_{K,b}$ is defined as $\mathsf{F}_K(\cdot)$ if $b = 0$, otherwise it is defined as a random function from $\{0,1\}^\lambda \to \{0,1\}^\lambda$.

Crypto Dark Matter PRF. Boneh, et al. [BIP+18] gave a novel strong PRF candidate which can be computed via a depth-3 $\mathsf{ACC}^0[p,q]$ circuit for primes $p < q$. As stated in the overview, this bypasses the supposed PRF barrier [BBKK18] and gives a simple, low-depth PRF $\mathsf{F} : \mathbb{Z}_p^{n' \times \ell m'} \times \mathbb{Z}_p^m \to \mathbb{Z}_q^n$, for $\ell = \left\lceil \log_p(q) \right\rceil$, defined as follows:

$$\mathsf{F}_{\mathbf{K}}(\mathbf{x}) := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1}\left(\mathbf{Y}_0 \cdot \mathbf{x} \bmod q\right) \bmod p\right) \bmod q \;,$$

where $\mathbf{K} \in \mathbb{Z}_p^{n' \times m'}$ is the secret PRF key, $\mathbf{x} \in \mathbb{Z}_p^m$ is the input, $\mathbf{Y}_0 \in \mathbb{Z}_q^{m' \times m}$ and $\mathbf{Y}_1 \in \mathbb{Z}_q^{n \times n'}$ $(n < n')$ are fixed public matrices and $\mathbf{G}^{-1}$ is the decomposition gadget which maps $\mathbb{Z}_q^{m'} \to \mathbb{Z}_q^{\ell m'}$.

Security. The security of the PRF is known to degrade when the key $\mathbf{K}$ is chosen to be a circulant matrix (instead of random) [CCKK22]. Aside from this, the scheme has held up to the recent cryptanalytic scrutiny [BIP+18, DGH+21] where the best attacks require exponential time and memory.

[7] For simplicity, we fix the key space, input space, and output space to be λ-bit strings, but this is not essential to the definition.

3.5 Public key encryption

A public key encryption (PKE) scheme PKE consists of the following polynomial-time algorithms.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$. The setup algorithm takes as input the security parameter λ, and outputs a public-secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, m) \to c$. The encryption algorithm takes as input a public key pk and a message m, and outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{sk}, c) \to m$. The decryption algorithm takes as input a secret key sk and a ciphertext c, and outputs a message m.

The scheme satisfies correctness if for all λ, m, $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$, and every ciphertext $c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m)$, we have that $\mathsf{Dec}(\mathsf{sk}, c) = m$. Further, the standard IND-CPA security notion is defined as follows.

Definition 3.6 (IND-CPA)**.** *A public key encryption scheme* PKE *is IND-CPA secure if for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible functions* $\epsilon(\cdot)$*, such that for every* $\lambda \in \mathbb{N}$

$$\Pr\left[\mathcal{A}(c) = b : \begin{array}{r} (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda) \\ b \leftarrow\!\!\$\ \{0,1\} \\ (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\ c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m_b) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda).$$

3.6 Non-interactive zero-knowledge

A non-interactive zero-knowledge (NIZK) proof system NIZK for relation $\mathscr{R}$ consists of the following polynomial time algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$. The setup algorithm takes as input the security parameter λ, and outputs a common reference string crs.
- $\mathsf{Prove}(\mathsf{crs}, \mathfrak{x}, \mathfrak{w}) \to \pi$. The prover algorithm takes as input the crs, an instance $\mathfrak{x} \in \mathscr{L}_{\mathscr{R}}$, and a witness $\mathfrak{w}$. It outputs a proof π.
- $\mathsf{Verify}(\mathsf{crs}, \mathfrak{x}, \pi) \to 0/1$. The verification algorithm takes as input the crs, an instance $\mathfrak{x}$, and a proof π. It outputs either 1 (accept) or 0 (reject).

Definition 3.7 (Non-interactive zero-knowledge)**.** *A NIZK proof system* NIZK *must satisfy the following properties:*

i. **Completeness.** *For every* $\lambda \in \mathbb{N}$*, crs* $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$*, any instance* $\mathfrak{x} \in \mathscr{L}_{\mathscr{R}}$ *with corresponding witness* $\mathfrak{w}$*,*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \mathfrak{x}, \pi) = 1 \ : \ \pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \mathfrak{x}, \mathfrak{w})\right] = 1.$$

ii. **Soundness.** *For every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{crs}, \mathfrak{x}, \pi) = 1 \\ \nexists \mathfrak{w} \ : \ (\mathfrak{x}, \mathfrak{w}) \in \mathscr{R} \end{array} \ : \ \begin{array}{r} \mathsf{crs} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda) \\ (\mathfrak{x}, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ .$$

iii. **(Multi-theorem) zero-knowledge.** *There exists a stateful* PPT *simulator* $\mathcal{S}$ *such that for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathcal{A}\left(\{\pi_{i,b}\}_i\right) = b \\ \wedge\ \forall i \in [\ell]\ :\ (\mathfrak{x}_i, \mathfrak{w}_i) \in \mathscr{R} \end{array} \ :\ \begin{array}{r} b \leftarrow\!\$\ \{0,1\} \\ \mathsf{crs}_0 \leftarrow\!\$\ \mathsf{Setup}(1^\lambda) \\ \mathsf{crs}_1 \leftarrow\!\$\ \mathcal{S}(1^\lambda) \\ \{\mathfrak{x}_i, \mathfrak{w}_i\}_{i\in[\ell]} \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}_b) \\ \forall i \in [\ell]\ :\ \pi_{i,0} \leftarrow\!\$\ \mathsf{Prove}(\mathsf{crs}_0, \mathfrak{x}_i, \mathfrak{w}_i) \\ \{\pi_{i,1}\}_i \leftarrow\!\$\ \mathcal{S}(\mathsf{crs}_1, \{\mathfrak{x}_i\}_i) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda)\ .$$

Argument of knowledge. *A NIZK proof system* NIZK *is an argument of knowledge (NIZKAoK) if it additionally satisfies the following extraction property:*

iv. **Knowledge extraction.** *There exists a* PPT *extractor* $\mathcal{E}$ *such that for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{crs}, \mathfrak{x}, \pi) = 1 \\ \wedge\ (\mathfrak{x}, \mathfrak{w}) \notin \mathscr{R} \end{array} \ :\ \begin{array}{r} \tau \leftarrow\!\$\ \{0,1\}^\lambda \\ \mathsf{crs} \leftarrow \mathcal{E}(1^\lambda; \tau) \\ (\mathfrak{x}, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}) \\ \mathfrak{w} \leftarrow \mathcal{E}(\tau, \mathfrak{x}, \pi) \end{array}\right] \leqslant \mathsf{negl}(\lambda)\ ,$$

where τ *denotes the randomness used for running the setup algorithm with* $\mathcal{E}$*, and we assume without any loss of generality that* $|\tau| = \lambda$*.*

Straight-line knowledge extraction. In our security proofs, we make regular use of straight-line extraction (i.e., a non-rewinding extractor). These can be constructed generically using PKE by encrypting the witness under a shared public encryption key and sending the ciphertext along with the NIZK proof. Importantly, the NIZK proof itself must contain a proof of well-formedness of the ciphertext.

3.7 Non-interactive Blind Signatures

In a non-interactive blind signature scheme, a signer issues a random *pre-signature* psig to any receiver with public key pk_R, so that the receiver can extract a blind signature σ for a random message μ using its secret key sk_R. Formally, a NIBS consists of the following polynomial-time algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$. On input the security parameter λ, the global setup algorithm outputs a set of public parameters pp. All algorithms take pp as an input, but we usually omit it as an explicit input for brevity.
- $\mathsf{KeyGen}_S(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{vk})$. On input pp, the signer's key generation algorithm samples a public (verification)-secret key pair $(\mathsf{sk}, \mathsf{vk})$.
- $\mathsf{KeyGen}_R(\mathsf{pp}) \to (\mathsf{sk}_R, \mathsf{pk}_R)$. On input pp, the receiver's key generation algorithm samples a public-secret key pair $(\mathsf{sk}_R, \mathsf{pk}_R)$.

- $\mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R) \to (\mathsf{psig}, \mathsf{nonce})$. The signer's issuance algorithm takes as input the signer's secret key sk as well as a receiver's public key pk_R. It then outputs a pre-signature psig along with nonce which represents (a portion of) the signer's random coins.
- $\mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, (\mathsf{psig}, \mathsf{nonce})) \to (\mu, \sigma)/\perp$. The receiver's blind signature extraction algorithm takes as input the receiver's secret key sk_R as an input, along with a verification key vk and pre-signature-nonce pair $(\mathsf{psig}, \mathsf{nonce})$, and outputs a message-signature pair (μ, σ) or aborts and outputs $\perp$.
- $\mathsf{Verify}(\mathsf{vk}, \mu, \sigma) \to 0/1$. The signature verification algorithm that takes as input a verification key vk and message-signature pair (μ, σ), and outputs 1 (accept) or 0 (reject).

Furthermore, a NIBS scheme NIBS must satisfy the following properties:

i. **Correctness.** For every security parameter $\lambda \in \mathbb{N}$, $\mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\ \mathsf{KeyGen}_S(\mathsf{pp})$, $(\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow\!\$\ \mathsf{KeyGen}_R(\mathsf{pp})$,

$$\Pr[\mathsf{Verify}(\mathsf{vk}, \mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R))) = 1] = 1 \ ,$$

where the probability is taken over the randomness of Issue and Obtain.

ii. **Reusability.** There exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l} \mathsf{nonce}_0 = \mathsf{nonce}_1 \\ \vee\ \mu_0 = \mu_1 \end{array} : \begin{array}{r} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\ \mathsf{KeyGen}_S(\mathsf{pp}) \\ (\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow\!\$\ \mathsf{KeyGen}_R(\mathsf{pp}) \\ \forall b \in \{0,1\}\ : \\ (\mathsf{psig}_b, \mathsf{nonce}_b) \leftarrow\!\$\ \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R) \\ (\mu_b, \sigma_b) \leftarrow\!\$\ \mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ .$$

iii. **One-more unforgeability.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l} \bigwedge_{i \in [Q+1]} \mathsf{Verify}(\mathsf{vk}, \mu_i, \sigma_i) = 1 \\ \bigwedge_{i \neq j \in [Q+1]} \mu_i \neq \mu_j \end{array} : \begin{array}{r} \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\ \mathsf{KeyGen}_S(\mathsf{pp}) \\ \{(\mu_i, \sigma_i)\}_{i=1}^{Q+1} \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{vk}) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ ,$$

where oracle $\mathcal{O}_{\mathsf{sk}}$ takes as input a receiver's public key $\mathsf{pk}_{R^{(i)}}$, and outputs a pre-signature-nonce pair $(\mathsf{psig}^{(i)}, \mathsf{nonce}^{(i)})$ by running $\mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(i)}})$, and $\mathcal{A}$ is *admissible* iff $\mathcal{A}$ makes at most Q queries to $\mathcal{O}_{\mathsf{sk}}$.

iv. **Receiver blindness.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l} \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}(\cdot,\cdot,\cdot)}(\mu_{\hat{b}}, \sigma_{\hat{b}}, \mu_{1-\hat{b}}, \sigma_{1-\hat{b}}) = \hat{b} : \\ \qquad \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda),\ \hat{b} \leftarrow\!\$\ \{0,1\}, \\ \qquad \forall b \in \{0,1\} :\ (\mathsf{sk}_{R_b}, \mathsf{pk}_{R_b}) \leftarrow\!\$\ \mathsf{KeyGen}_R(\mathsf{pp}) \\ \qquad (\mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)_{b \in \{0,1\}}) \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}(\cdot,\cdot,\cdot)}(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}) \\ \qquad \forall b \in \{0,1\}\ :\ (\mu_b, \sigma_b) \leftarrow\!\$\ \mathsf{Obtain}(\mathsf{sk}_{R_b}, \mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda) \ ,$$

where oracle $\mathcal{O}_{\mathsf{sk}_{R_0},\mathsf{sk}_{R_1}}$, on the ith query $\left(b^{(i)},\mathsf{vk}^{(i)},\left(\mathsf{psig}^{(i)},\mathsf{nonce}^{(i)}\right)\right)$, outputs $\mathsf{Obtain}(\mathsf{sk}_{R_{b^{(i)}}}, \mathsf{vk}^{(i)},(\mathsf{psig}^{(i)},\mathsf{nonce}^{(i)}))$. That is, $\mathcal{O}_{\mathsf{sk}_{R_0},\mathsf{sk}_{R_1}}$ provides $\mathcal{A}$ oracle access to the Obtain algorithm w.r.t. $\mathsf{sk}_{R_0},\mathsf{sk}_{R_1}$. We say that $\mathcal{A}$ is admissible iff:

a. $\sigma_0,\sigma_1 \neq \perp$ (i.e., Obtain algorithm does not abort), and
b. $\mathsf{nonce}_0 \neq \mathsf{nonce}^{(i)}$ and $\mathsf{nonce}_1 \neq \mathsf{nonce}^{(i)}$ for all i. (That is, $\mathcal{A}$ cannot make an Obtain query with nonce value to be either of the challenge nonce values.)

v. **Nonce blindness.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{r} \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_R}(\cdot,\cdot)}(\mu_{\hat{b}},\sigma_{\hat{b}},\mu_{1-\hat{b}},\sigma_{1-\hat{b}}) = \hat{b} : \\ \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda),\ (\mathsf{sk}_R,\mathsf{pk}_R) \leftarrow \mathsf{KeyGen}_R(\mathsf{pp}) \\ (\mathsf{vk},(\mathsf{psig}_b,\mathsf{nonce}_b)_{b\in\{0,1\}}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_R}(\cdot,\cdot)}(\mathsf{pk}_R),\ \hat{b} \leftarrow \{0,1\} \\ \forall b \in \{0,1\}\ :\ (\mu_b,\sigma_b) \leftarrow \mathsf{Obtain}(\mathsf{sk}_R,\mathsf{vk},(\mathsf{psig}_b,\mathsf{nonce}_b)) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda)\ ,$$

where oracle $\mathcal{O}_{\mathsf{sk}_R}$, on the ith query $\left(\mathsf{vk}^{(i)},\left(\mathsf{psig}^{(i)},\mathsf{nonce}^{(i)}\right)\right)$, outputs $\mathsf{Obtain}(\mathsf{sk}_R,\mathsf{vk}^{(i)},(\mathsf{psig}^{(i)},\mathsf{nonce}^{(i)}))$. That is, $\mathcal{O}_{\mathsf{sk}_R}$ provides $\mathcal{A}$ oracle access to the Obtain algorithm w.r.t. sk_R. We say that $\mathcal{A}$ is admissible iff:

a. $\sigma_0,\sigma_1 \neq \perp$ (i.e., Obtain algorithm does not abort), and
b. $\mathsf{nonce}_0 \neq \mathsf{nonce}^{(i)}$ and $\mathsf{nonce}_1 \neq \mathsf{nonce}^{(i)}$ for all i. (That is, $\mathcal{A}$ cannot make an Obtain query with nonce value to be either of the challenge nonce values.)

4 Non-interactive Batched Blind Signatures

A non-interactive *batched* blind signature (NIBBS) scheme extends a standard NIBS scheme so that, given a single random pre-signature psig, a receiver with public key pk_R can extract $N \in \mathbb{N}$ blind signatures $\sigma_1,\sigma_2,\ldots,\sigma_N$ for the corresponding random messages $\mu_1,\mu_2,\ldots,\mu_N$ using its secret key sk_R. Formally, a NIBBS consists of the following polynomial-time algorithms:

- $\mathsf{Setup}(1^\lambda,N) \to \mathsf{pp}$. On input the security parameter λ and a batch size N, the global setup algorithm outputs a set of public parameters pp. All algorithms take pp as an input, but we usually omit it as an explicit input for brevity.

- $\mathsf{KeyGen}_S(\mathsf{pp}) \to (\mathsf{sk},\mathsf{vk})$. On input pp, the signer's key generation algorithm samples a public (verification)-secret key pair $(\mathsf{sk},\mathsf{vk})$.

- $\mathsf{KeyGen}_R(\mathsf{pp}) \to (\mathsf{sk}_R,\mathsf{pk}_R)$. On input pp, the receiver's key generation algorithm samples a public-secret key pair $(\mathsf{sk}_R,\mathsf{pk}_R)$.

- $\mathsf{Issue}(\mathsf{sk},\mathsf{pk}_R) \to (\mathsf{psig},\mathsf{nonce})$. The signer's issuance algorithm takes as input the signer's secret key sk as well as a receiver's public key pk_R. It then outputs a pre-signature psig along with nonce which represents (a portion of) the signer's random coins.

- $\mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, (\mathsf{psig}, \mathsf{nonce})) \to [\![\mu, \sigma]\!]/\perp$. The receiver's blind signature extraction algorithm takes as input the receiver's secret key sk_R, along with a verification key vk and pre-signature-nonce pair $(\mathsf{psig}, \mathsf{nonce})$, and outputs N message-signature pairs $[\![\mu, \sigma]\!] := \{(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_N, \sigma_N)\}$ or aborts and outputs $\perp$.

- $\mathsf{Verify}(\mathsf{vk}, \mu, \sigma) \to 0/1$. The signature verification algorithm that takes as input a verification key vk and message-signature pair (μ, σ), and outputs 1 (accept) or 0 (reject).

Furthermore, in addition to the standard notion of correctness, a NIBS scheme with batching must satisfy the following properties:

i. **Succinct communication.** For $\lambda \in \mathbb{N}$, there exists a polynomial $\mathsf{poly}(\lambda)$ such that $|\mathsf{psig}|, |\mathsf{nonce}| \leqslant \mathsf{poly}(\lambda)$.

ii. **Reusability.** There exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l|r} \mathsf{nonce}_0 = \mathsf{nonce}_1 \\ \bigvee_{\mathrm{i,j} \in [N]} \mu_{\mathrm{i},0} = \mu_{\mathrm{j},1} \end{array} : \begin{array}{r} \mathsf{pp} \leftarrow\!\$\, \mathsf{Setup}(1^\lambda) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\, \mathsf{KeyGen}_S(\mathsf{pp}) \\ (\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow\!\$\, \mathsf{KeyGen}_R(\mathsf{pp}) \\ \forall b \in \{0,1\} : \\ (\mathsf{psig}_b, \mathsf{nonce}_b) \leftarrow\!\$\, \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R) \\ [\![\mu, \sigma]\!]_b \leftarrow\!\$\, \mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ .$$

iii. **One-more unforgeability.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l|r} \bigwedge_{i \in [NQ+1]} \mathsf{Verify}(\mathsf{vk}, \mu_i, \sigma_i) = 1 \\ \bigwedge_{i \neq j \in [NQ+1]} \mu_i \neq \mu_j \end{array} : \begin{array}{r} \mathsf{pp} \leftarrow\!\$\, \mathsf{Setup}(1^\lambda) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\, \mathsf{KeyGen}_S(\mathsf{pp}) \\ \{(\mu_i, \sigma_i)\}_{i=1}^{NQ+1} \leftarrow\!\$\, \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{vk}) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ ,$$

where oracle $\mathcal{O}_{\mathsf{sk}}$ takes as input a receiver's public key $\mathsf{pk}_{R^{(j)}}$, and outputs a pre-signature-nonce pair $(\mathsf{psig}^{(j)}, \mathsf{nonce}^{(j)})$ by running $\mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(j)}})$, and $\mathcal{A}$ is *admissible* iff $\mathcal{A}$ makes at most Q queries to $\mathcal{O}_{\mathsf{sk}}$.

iv. **Inter-batch blindness.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{r} \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}(\cdot,\cdot,\cdot)}([\![\mu, \sigma]\!]_{\hat{b}}, [\![\mu, \sigma]\!]_{1-\hat{b}}) = \hat{b} : \\ \mathsf{pp} \leftarrow\!\$\, \mathsf{Setup}(1^\lambda),\ \hat{b} \leftarrow\!\$\, \{0,1\}, \\ \forall b \in \{0,1\} :\ (\mathsf{sk}_{R_b}, \mathsf{pk}_{R_b}) \leftarrow\!\$\, \mathsf{KeyGen}_R(\mathsf{pp}) \\ (\mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)_{b \in \{0,1\}}) \leftarrow\!\$\, \mathcal{A}^{\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}(\cdot,\cdot,\cdot)}(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}) \\ \forall b \in \{0,1\} :\ [\![\mu, \sigma]\!]_b \leftarrow\!\$\, \mathsf{Obtain}(\mathsf{sk}_{R_b}, \mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda) \ ,$$

where oracle $\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}$, on the jth query $\left(b^{(j)}, \mathsf{vk}^{(j)}, \left(\mathsf{psig}^{(j)}, \mathsf{nonce}^{(j)}\right)\right)$, outputs $\mathsf{Obtain}(\mathsf{sk}_{R_{b^{(j)}}}, \mathsf{vk}^{(j)}, (\mathsf{psig}^{(j)}, \mathsf{nonce}^{(j)}))$. That is, $\mathcal{O}_{\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}}$ provides $\mathcal{A}$ oracle access to the Obtain algorithm w.r.t. $\mathsf{sk}_{R_0}, \mathsf{sk}_{R_1}$. We say that $\mathcal{A}$ is admissible iff:

a. For all $\mathrm{i} \in [N]$ and $b \in \{0,1\}$, $\sigma_{\mathrm{i},b} \neq \perp$ (i.e., Obtain algorithm does not abort), and

b. $\mathsf{nonce}_0 \neq \mathsf{nonce}^{(j)}$ and $\mathsf{nonce}_1 \neq \mathsf{nonce}^{(j)}$ for all j. (That is, $\mathcal{A}$ cannot make an Obtain query with nonce value to be either of the challenge nonce values.)

v. **Intra-batch blindness.** For every *stateful*, *admissible* PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{c} \mathcal{A}^{O_{\mathsf{sk}_R}(\cdot,\cdot)}\left(\left\{\left(\mu_{\varphi_{\hat{b}}(i,b)}, \sigma_{\varphi_{\hat{b}}(i,b)}\right)\right\}_{\substack{i\in[N]\\ b\in\{0,1\}}}\right) = \hat{b} : \\ \mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda),\ (\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow\$ \mathsf{KeyGen}_R(\mathsf{pp}), \hat{b} \leftarrow\$ \{0,1\} \\ \varphi_0 := \iota,\ (\mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)_{b\in\{0,1\}}, \varphi_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}_R}(\cdot,\cdot)}(\mathsf{pk}_R) \\ \forall b \in \{0,1\} \ :\ [\![\mu, \sigma]\!]_b \leftarrow\$ \mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, (\mathsf{psig}_b, \mathsf{nonce}_b)) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda)\ ,$$

where φ_0 and φ_1 are bijective permutation maps in $[N]\times\{0,1\} \to [N]\times\{0,1\}$ with φ_0, in particular, being the identity map ι; and oracle O_{sk_R}, on the jth query $\left(\mathsf{vk}^{(j)}, \left(\mathsf{psig}^{(j)}, \mathsf{nonce}^{(j)}\right)\right)$, outputs $\mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}^{(j)}, (\mathsf{psig}^{(j)}, \mathsf{nonce}^{(j)}))$. That is, O_{sk_R} provides $\mathcal{A}$ oracle access to the Obtain algorithm w.r.t. sk_R. We say that $\mathcal{A}$ is admissible iff:

a. φ_1 is a valid permutation map in in $[N]\times\{0,1\} \to [N]\times\{0,1\}$.

b. For all $\mathrm{i} \in [N]$ and $b \in \{0,1\}$, $\sigma_{\mathrm{i},b} \neq \perp$ (i.e., Obtain algorithm does not abort), and

c. $\mathsf{nonce}_0 \neq \mathsf{nonce}^{(j)}$ and $\mathsf{nonce}_1 \neq \mathsf{nonce}^{(j)}$ for all j. (That is, $\mathcal{A}$ cannot make an Obtain query with nonce value to be either of the challenge nonce values.)

5 Lattice-based NIBBS

Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}, \mathfrak{B}_N = (\mathfrak{s} + n'm_N)\sqrt{m}$, and positive integer q be functions of the security parameter λ such that the randomized one-more ISIS instance $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ is hard. These parameters must satisfy the following constraints:

$$n = \mathsf{poly}(\lambda),\ n' > n,\ m > n\log q + \Theta(\lambda),\ n'm_N > n\log q + \Theta(\lambda)\,. \tag{1}$$

This construction relies on a hash function $\mathsf{H} : \mathbb{Z}_2^\lambda \to \mathbb{Z}_q^n$ (modeled as a random oracle), a lattice trapdoor $\mathcal{T} = (\mathcal{T}.\mathsf{TrapGen}, \mathcal{T}.\mathsf{SamplePre})$, a public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ and a NIZK proof system $\mathsf{NIZK} = (\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ for relation $\mathscr{R}^{\mathsf{NIBBS}}$, defined as follows:

Relation $\mathscr{R}^{\mathsf{NI\underline{B}BS}}$

Instance: Each instance $\mathfrak{x}$ is interpreted as matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0$ and $\mathbf{Y}_1$, a PKE public key $\mathsf{pk}_{\mathsf{PKE}}$, a message vector μ and ciphertext ψ.

Witness: Witness $\mathfrak{w}$ consists of secret vectors $\mathbf{k}, \mathbf{u}, \mathbf{s}, \mathbf{r}$ and $\mathbf{i}$, and PKE randomness ρ.

Membership: $\mathfrak{w}$ is a valid witness for $\mathfrak{x}$ if the following are satisfied:

- $\mathbf{A} \cdot \mathbf{s} + \mathbf{B} \cdot \mathbf{r} = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$
- $\mu = \mathbf{Y}_1 \cdot \left(\mathsf{Vec}^{-1}(\mathbf{k}) \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r} \\ \mathbf{i} \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3$
- $\psi = \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u} \| \mathbf{s} \| \mathbf{r} \| \mathbf{i} \; ; \; \rho)$
- $\|\mathbf{s}\| \leqslant \mathfrak{B}, \quad \mathbf{k} \in \mathbb{Z}_2^{n' m_N}, \quad \mathbf{r} \in \mathbb{Z}_2^{m}, \quad \mathbf{i} \in \mathbb{Z}_2^{m_N - m'}$.

5.1 Construction

We are now ready to describe our lattice-based NI$\underline{\mathsf{B}}$BS scheme.

- $\mathsf{Setup}(1^\lambda, N) \to \mathsf{pp}$. The public parameter setup algorithm computes $n, m, n', m', \mathfrak{s}, q$ from the security parameter λ. It then sets $m_N = m' + \lceil \log_2(N) \rceil$, and samples matrices $\mathbf{Y}_0 \leftarrow\!\!\$\; \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\!\$\; \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\!\$\; \mathbb{Z}_q^{n \times n' m_N}$. It then runs the key generation algorithm of PKE and generates the corresponding public and secret key pair as

$$(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\; \mathsf{PKE.KeyGen}(1^\lambda) \, .$$

  Next, it generates $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\!\$\; \mathsf{NIZK.Setup}(1^\lambda)$ and outputs

$$\mathsf{pp} := \left(1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}\right) .$$

  The public parameters are fixed once and for all.

- $\mathsf{KeyGen}_S(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{vk})$. The signer's key generation algorithm runs the lattice trapdoor setup to obtain

$$(\mathbf{T}_{\mathbf{A}}, \mathbf{A}) \leftarrow\!\!\$\; \mathcal{T}.\mathsf{TrapGen}(1^\lambda, n, m, q) \, .$$

  It also samples a matrix $\mathbf{B} \leftarrow\!\!\$\; \mathbb{Z}_q^{n \times m}$, then outputs the signer's secret signing key $\mathsf{sk} := \mathbf{T}_{\mathbf{A}}$ and verification key $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$.

- $\mathsf{KeyGen}_R(\mathsf{pp}) \to (\mathsf{sk}_R, \mathsf{pk}_R)$. The receiver's key generation algorithm samples the PRF key $\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n' \times m_N}$ and a random vector $\mathbf{u} \leftarrow\!\!\$\ \mathbb{Z}_2^{\lambda}$. It then vectorizes $\mathbf{K}$ as $\mathbf{k} \coloneqq \mathsf{Vec}(\mathbf{K})$, and computes

$$\mathbf{t} \coloneqq \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u}) \,.$$

  Finally, it sets the receiver's secret key as $\mathsf{sk}_R \coloneqq (\mathbf{K}, \mathbf{u})$ and the public key as $\mathsf{pk}_R \coloneqq \mathbf{t}$, and outputs them.

- $\mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R) \to (\mathsf{psig}, \mathsf{nonce})$. Given the receiver's public key $\mathsf{pk}_R =: \mathbf{t}$, the signer's pre-signature issuance algorithm samples a random vector $\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_2^m$ and, using the secret signing key $\mathsf{sk} =: \mathbf{T}_\mathbf{A}$, it computes

$$\mathbf{s} \leftarrow\!\!\$\ \mathcal{T}.\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{t} - \mathbf{B} \cdot \mathbf{r}, \mathfrak{s}),$$

  and outputs the pre-signature, $\mathsf{psig} \coloneqq \mathbf{s}$ and the nonce, $\mathsf{nonce} \coloneqq \mathbf{r}$.

- $\mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, \mathsf{psig}, \mathsf{nonce}) \to [\![\mu, \sigma]\!]$. The receiver's signature obtainment algorithm first parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, $\mathsf{psig} =: \mathbf{s}$, and $\mathsf{nonce} =: \mathbf{r}$, it checks if

$$\mathbf{A} \cdot \mathbf{s} + \mathbf{B} \cdot \mathbf{r} \stackrel{?}{=} \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}) + \mathsf{H}(\mathbf{u}) \ \wedge\ \|\mathbf{s}\| \stackrel{?}{\leqslant} \mathfrak{B} \ \wedge\ \mathbf{r} \stackrel{?}{\in} \mathbb{Z}_2^m \,.$$

  If any check fails it aborts and outputs $\perp$. Otherwise, for each $\mathrm{i} \in [N]$, it computes the message μ_i as

$$\mu_\mathrm{i} \coloneqq \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r} \\ \mathbf{g}^{-1}(\mathrm{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,,$$

  where, $\mathbf{g}^{-1} : \mathbb{Z}_N \to \mathbb{Z}_2^{\log_2(N)}$ ($\log_2(N) = m_N - m'$) is the binary decomposition gadget.
  With this, it now generates the corresponding ciphertext as

$$\psi_\mathrm{i} \leftarrow \mathsf{PKE.Enc}\left(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u} \| \mathbf{s} \| \mathbf{r} \| \mathbf{g}^{-1}(\mathrm{i}) \,;\, \rho_\mathrm{i}\right)$$

  from uniformly sampled randomness ρ_i. It then generates the NIZK proof

$$\pi_\mathrm{i} \leftarrow\!\!\$\ \mathsf{NIZK.Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_\mathrm{i} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_\mathrm{i}, \psi_\mathrm{i}), \\ \mathfrak{w}_\mathrm{i} \coloneqq \left(\mathbf{k}, \mathbf{u}, \mathbf{s}, \mathbf{r}, \mathbf{g}^{-1}(\mathrm{i}), \rho_\mathrm{i}\right) \end{array} \right),$$

  Finally, it outputs $[\![\mu, \sigma]\!] \coloneqq \{(\mu_1, \sigma_1 \coloneqq (\psi_1, \pi_1)), \ldots, (\mu_N, \sigma_N \coloneqq (\psi_N, \pi_N))\}$.

- $\mathsf{Verify}(\mathsf{vk}, \mu, \sigma) \to 0/1$. The signature verification algorithm parses σ as (ψ, π) and outputs 1 (accept) if

$$\mathsf{NIZK.Verify}(\mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu, \psi), \pi) = 1 \,,$$

  and 0 (reject) otherwise.

Security. We now state our main theorem concerning the security of our NIBBS scheme.

Theorem 5.1. *Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}, \mathfrak{B}_N = (\mathfrak{s} + n'm_N)\sqrt{m}$, and positive integer q be functions of the security parameter λ satisfying (1) such that the randomized one-more ISIS instance $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ is hard.*

If $\mathsf{F}_{\mathbf{K}}$ is a pseudorandom function, NIZK is a NIZK for $\mathscr{R}^{\mathsf{NIBBS}}$ and PKE is an IND-CPA secure encryption scheme then construction 5 is a secure non-interactive batched blind signature scheme.

Proof of theorem. We prove the correctness, succinctness (of communication), reusability, one-more unforgeability, and inter- and intra-batch blindness of our scheme. The theorem will accordingly follow.

Correctness. By correctness of trapdoor sampling, $\mathbf{s}$ is a valid preimage of $(\mathbf{t} - \mathbf{B} \cdot \mathbf{r})$ with respect to $\mathbf{A}$ such that $\|\mathbf{s}\| \leqslant \mathfrak{B}$ and $\mathbf{r} \in \mathbb{Z}_2^m$. Since it further holds, by construction, that $\mathsf{Vec}(\mathbf{K}) \in \mathbb{Z}_2^{n'm_N}$, and for each $\mathsf{i} \in [N]$, $\mu_{\mathsf{i}} = \mathsf{F}_{\mathbf{K}}(\mathbf{r} \| \mathbf{g}^{-1}(\mathsf{i}))$, it follows from correctness of PKE and completeness of NIZK, that each π_{i} is a valid proof for $\mathscr{R}^{\mathsf{NIBBS}}$. Consequently, construction 5 is correct.

Succinct communication. This follows straightforwardly from the construction as the size of both $\mathbf{s} \in \mathbb{Z}^m$ and $\mathbf{r} \in \mathbb{Z}_2^m$ is independent of N.

Next, we consider the following lemmas concerning the reusability, one-more unforgeability, as well as inter- and intra-batch blindness of the protocol. We provide the detailed proofs in Appendix A.

Lemma 5.2 (Reusability)**.** *Suppose $\mathsf{F}_{\mathbf{K}}$ is a pseudorandom function. Then, construction 5 is reusable.*

Proof. For $b \in \{0,1\}$, let $(\mathbf{s}_{(b)}, \mathbf{r}_{(b)})$ be the pre-signature and nonce pairs given to a receiver with $\mathsf{pk}_R := \mathbf{t}$, then one can check that

$$\Pr\left[\mathbf{r}_{(0)} = \mathbf{r}_{(1)} \vee \left(\bigvee_{\mathsf{i},\mathsf{j} \in [B]} \mu_{\mathsf{i},0} = \mu_{\mathsf{j},1}\right)\right] \leqslant$$
$$\sum_{\mathsf{i},\mathsf{j} \in [N]} \Pr\left[\mathsf{F}_{\mathbf{K}}\left(\begin{bmatrix}\mathbf{r}_{(0)} \\ \mathbf{g}^{-1}(\mathsf{i})\end{bmatrix}\right) = \mathsf{F}_{\mathbf{K}}\left(\begin{bmatrix}\mathbf{r}_{(1)} \\ \mathbf{g}^{-1}(\mathsf{j})\end{bmatrix}\right)\right] + \mathsf{negl}(\lambda)\,.$$

Now, if with non-negligible probability, $\mathsf{F}_{\mathbf{K}}\left(\begin{bmatrix}\mathbf{r}_{(0)} \\ \mathbf{g}^{-1}(\mathsf{i})\end{bmatrix}\right) = \mathsf{F}_{\mathbf{K}}\left(\begin{bmatrix}\mathbf{r}_{(1)} \\ \mathbf{g}^{-1}(\mathsf{j})\end{bmatrix}\right)$ for any $\mathsf{i}, \mathsf{j} \in [N]$ and for uniform choice of $\mathbf{r}_{(b)}$, then clearly $\mathsf{F}_{\mathbf{K}}$ is distinguishable from a random function, which would contradict our assumption. It follows that the above probability must be negligible, and consequently the construction is reusable. □

Lemma 5.3 (One-more unforgeability)**.** *Suppose the $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ problem is hard and NIZK is sound. Then, construction 5 is one-more unforgeable.*

Pf. sketch. The proof of one-more unforgeability follows via a reduction to the rOM-ISIS assumption. The reduction sets $\mathbf{C} := \mathbf{A}_\top \cdot \mathbf{R}$ for $\mathbf{R}$ a uniformly chosen binary matrix. The fact that this change is indistinguishable to an adversary can be shown via the leftover hash lemma (Corollary 3.2). Then, whenever the adversary makes a query for the hash of some value $\mathbf{u}$, the reduction

queries the rOM-ISIS oracle for a syndrome $\mathbf{h}$ and programs the random oracle such that $\mathsf{H}(\mathbf{u}) \coloneqq \mathbf{h}$. When the adversary outputs its (one-more) forgeries, the reduction extracts $(\mathbf{k}_j, \mathbf{u}_j, \mathbf{s}_j, \mathbf{r}_j)$ from each proof π_j. With high probability, the adversary must have queried each $\mathbf{u}_j$ to the random oracle such that $\mathbf{h}_j = \mathsf{H}(\mathbf{u}_j)$ so that each $\mathbf{h}_j$ is in the set $\mathscr{S}$. So that $\mathbf{h}_j = \mathbf{A} \cdot \left(\mathbf{s}_j - \mathbf{R} \cdot \mathbf{k}_i\right) + \mathbf{B} \cdot \mathbf{r}_j$. Consequently, at least one of $(\mathbf{s}_i - \mathbf{R} \cdot \mathbf{k}_i)$ is a valid rOM-ISIS solution. □

Lemma 5.4 (Inter-batch blindness)**.** *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK *is a multi-theorem zero-knowledge proof system and* PKE *is* IND-CPA *secure. Then, construction 5 is inter-batch blind.*

Pf. sketch. To prove batch receiver blindness, we will have the challenger simulate its NIZK proof, and then compute the encryptions ψ_{i}'s as encryptions of $\mathbf{0}$ for each receiver. These changes can be shown to be indistinguishable to an adversary by the zero-knowledge of the NIZK proof system, and IND-CPA security of the the public key encryption scheme respectively. Next, leveraging the leftover hash lemma (Corollary 3.2), we can have the challenger sample each receiver public key $\mathbf{t}$ uniformly. Notice, at this stage, that the adversary learns no information about the receiver and blindness follows. □

Lemma 5.5 (Intra-batch blindness)**.** *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK *is multi-theorem zero-knowledge proof system and* PKE *is* IND-CPA *secure. Then, construction 5 is intra-batch blind.*

Pf. sketch. The proof of batch nonce blindness is quite similar to the previous proof—the challenger simulates its NIZK proof, then computes the encryptions ψ_{i}'s as encryptions of $\mathbf{0}$, and then leveraging the leftover hash lemma (Corollary 3.2), the challenger samples each receiver public key $\mathbf{t}$ uniformly. Since the adversary learns no information about the receiver, blindness follows. □

5.2 Instantiation

In this section, we propose concrete parameters to instantiate our NIBBS protocol and provide rough estimates for the size of the final signature. In choosing our parameters, we aim for 128-bit classical security.

Preimage sampling paramters. We instantiate the trapdoor generation and preimage sampling procedures using NTRU lattices over the Falcon-512 ring $\mathcal{R}_{\mathsf{F}} \coloneqq \mathcal{R}_{q_{\mathsf{F}}, d_{\mathsf{F}}}$, where $q_{\mathsf{F}} = 12289$ and $d_{\mathsf{F}} = 512$. . The other parameters are computed as prescribed in [PFH+22, §2.6]). Concretely,

$$\mathfrak{s} = \frac{1.17}{\pi}\sqrt{q_{\mathsf{F}} \cdot \log\left(4d_{\mathsf{F}} \cdot \left(1 + \sqrt{\lambda \cdot 2^{64}}\right)/2\right)} \approx 2^{7.37} \text{ and } \mathfrak{B} = 1.1 \cdot \mathfrak{s}\sqrt{2d_{\mathsf{F}}}\,.$$

PRF parameters. We carry over the parameters from [ADDG24] for our desired security level so that $n = 82$ and $n', m_N = 256$ (so that $m' = m_N - \lceil \log_2(B) \rceil$), except we set $m = 1024$ for sufficient rOM-ISIS hardness. Note that, since the final message is in $\mathbb{Z}_3^n$, the final message space is sufficiently large ($2^{\Theta(\lambda)}$).

Receiver keys. Let $\mathsf{Coeff} : \mathbb{Z}[X]/(X^d + 1) \to \mathbb{Z}^d$ be the canonical coefficient map, and $\mathsf{Coeff}^{-1} : \mathbb{Z}^d \to \mathbb{Z}[X]/(X^d + 1)$ be its inverse. Then, the PRF key $\mathbf{K} \in \mathbb{Z}_2^{n' \times m_N}$ can be interpreted as $n' m_N / d_{\mathsf{F}} =$

128 polynomials $(k_1, \ldots, k_{128}) = \mathsf{Coeff}^{-1}(\mathsf{Vec}(\mathbf{K}))$. For $u \leftarrow\!\$\, \{0,1\}^{256}$, the receiver's public key $t \in \mathcal{R}_{\mathsf{F}}$ is then given as

$$t := \sum_{j=1}^{128} c_j \cdot k_j + \mathsf{H}(u) \bmod q_{\mathsf{F}} \,,$$

where $\mathsf{H} : \{0,1\}^{256} \to \mathcal{R}_{\mathsf{F}}$ is a secure hash function such as SHA-3-256 and $c_i \in \mathcal{R}_{\mathsf{F}}$ are sampled during protocol setup.

Pre-signature issuance. Given a receiver public key $t \in \mathcal{R}_{\mathsf{F}}$, the signer first samples the nonce vector $\mathbf{r} \leftarrow\!\$\, \mathbb{Z}_2^m$, and then decomposes it into $m/d = 2$ polynomials $(r_1, r_2) = \mathsf{Coeff}^{-1}(\mathbf{r})$. The signer then uses its NTRU secret trapdoor to sample short polynomials $s_1, s_2 \in \mathcal{R}_{\mathsf{F}}$ with respect to $a \in \mathcal{R}_{\mathsf{F}}$ such that

$$a \cdot s_1 + s_2 + b \cdot r_1 + r_2 = t \bmod q_{\mathsf{F}} \,,$$

for $b \in \mathcal{R}_{\mathsf{F}}$ a part of the signer's verification key. The presignature is then s_1 and the nonce is $\mathbf{r}$ (the receiver can recover s_2 on its own). An important change, following the template of [AKSY22] is that we use rejection sampling to keep the ℓ_∞ norm of (s_1, s_2) to be at most $\lceil 4.15 \cdot \mathfrak{s} \rceil = 688$ for compatibility with the encryption scheme. Using, this bound, we get that the size of the signer's message is at most 0.62 KB for the presignature and an additional 0.12 KB for the nonce.

Blind signature generation. The receiver encrypts 133 polynomials in $\mathcal{R}_{\mathsf{F}}$—$\left\{k_j\right\}_{j\in[128]}, r_1, r_2, s_1$, along with u and $\mathfrak{i}$ (as constant coefficients) using the Regev style public key encryption scheme from [LPS10] over $\mathcal{R}_{q_{\mathsf{PKE}}, d_{\mathsf{F}}}$. Thus, the rank of the plaintext space is 133, which is the main source of inefficiency in our final signature, as the required PKE modulus is $q_{\mathsf{PKE}} = 210596593 \approx 2^{27.6}$. Consequently, the ciphertext size is 231.53 KB, and can be brought down to around 214.78 KB by dropping lower order bits [SAB+22].

Unfortunately, the Falcon parameters do not satisfy the necessary conditions for the use of the modular Johnson-Lindenstrauss projections [BS23, Lemma 2.2] (also see [AAB+24]). So, we must lift the signature verification equation to a larger modulus ring $\mathcal{R}_{q_{\mathsf{L}}, d_{\mathsf{F}}}$ for modulus $q_{\mathsf{L}} > q_{\mathsf{F}}$. To simplify our final estimates, we set it to be as close as possible to the original LaBRADOR modulus 2^{32}. Now, we can lift the signature verification check directly to $\mathcal{R}$ (modulo nothing) so that

$$a \cdot s_1 + s_2 + b \cdot r_1 + r_2 + q_{\mathsf{F}}\delta = \sum_{j=1}^{128} c_j \cdot k_j + \mathsf{H}(u) \in \mathcal{R} \,, \tag{2}$$

where the modular wraparound $\delta \in \mathcal{R}_{q_{\mathsf{L}}, d_{\mathsf{F}}}$ must now be made part of the witness. For appropriately bounded δ, if (2) holds over $\mathcal{R}$, it must hold over $\mathcal{R}_{q_{\mathsf{L}}, d_{\mathsf{F}}}$.

Further, to be compatible with the degree of the polynomial-ring in LaBRADOR, i.e., $d_{\mathsf{L}} = 64$, we use a reduced degree polynomial-ring. Note that the constraints over the larger ring $\mathcal{R}_{\mathsf{F}}$ can be reformulated as constraints over any sub-ring of degree d/c for some constant c. Specifically, the encryption scheme is instantiated over $\mathcal{R}_{q_{\mathsf{F}}, d_{\mathsf{L}}}$ where $(\mathcal{R}_{q_{\mathsf{F}}, d_{\mathsf{L}}}^{d_{\mathsf{F}}/d_{\mathsf{L}}} =)\ \mathcal{R}_{q_{\mathsf{F}}, d_{\mathsf{L}}}^{8} \cong \mathcal{R}_{\mathsf{F}}$ and this bijection preserves the norms [LNPS21].

The receiver must prove knowledge of the polynomials satisfying the relation $\mathscr{R}^{\mathsf{NIBBS}}$ with respect to the instantiation discussed in this section. In total, we approximate that the overall relation involves $\approx 2^{19.5}$ R1CS constraints (Table 3), which gives a proof size of 58 KB using

LaBRADOR (cf. [BS23, § 7.1]). However, this only accounts for the SNARK without the additional zero-knowledge property. The LaBRADOR paper gives limited guidance on how to make the protocol zero-knowledge, but the main idea is to deploy a NIZK proof system at an intermediate level of the LaBRADOR recursion (so that the witness is already quite small). This involves some overhead required for blinding the outer commitments up to the NIZK level, masking the Johnson-Lindenstrauss projection and some additional garbage polynomials. In particular, for a binary-R1CS with 2^{20} constraints, using the lattice-based NIZK proof system for linear relations [LNP22b], the overhead is about 11 KB (cf. [BS23, § 6]). Since a single R1CS constraint can be expressed in no more than $\log_2^2(q_\mathsf{L}) \approx 2^{10}$ constraints, and given the slow order of growth of the LaBRADOR proofs, a very rough estimate puts the overhead for 2^{20} (general) R1CS constraints at about 36 KB, although with the right instantiation, we expect the actual overhead to be a few kilobytes lower. Nevertheless, with full zero-knowledge, the signature size is $58 + 36 + 214.78 = 308.78$ KB.

Component	**#Constraints**
Hash computation (SHA-3-256)	27000
Signature verification [PFH+22]	68609
PRF computation [BIP+18]	286000
PKE computation [LPS10]	273408
ℓ_2-norm bounds	73019
Binary coefficient checks	66560
Total	**794596**

Table 3: Approximate R1CS constraint count.

We sketch out our calculations of the number of constraints in the following section. These estimates are somewhat conservative, and further optimizations and reductions in the proof size may be possible. In particular, using SNARK-friendly hash functions [AGR+16] can also significantly reduce the number of constraints, leading to modest savings in the proof size.

Remark 5.6 (Instantiation of lattice-based NIBS). Our scheme can also readily be used as a standard (non-batched) NIBS. The main distinction is the absence of the index i which allows for a few minor optimizations. In particular, the signer can now precompute $\mathbf{G}^{-1}(\mathbf{Y}_0 \cdot \mathbf{r} \bmod 3)$ and send the resulting $\tilde{\mathbf{r}} \in \mathbb{Z}_3^{256}$. This increases the size of the signer's message by just 0.06 KB. On the other hand, it reduces the receiver's computation time, as well as the final signature size by 2.6 KB (1.6 KB ciphertext and 1 KB proof) to 306.18 KB after compression. Furthermore, and as we show in Section 6, one can entirely do away with the hash computation if one is willing to rely on the recently proposed ISIS_f assumption [BLNS23b].

5.3 Estimating R1CS constraint count

In this section, we sketch out our computation of the number of R1CS constraints shown in Table 3.

Hash computation. Standard SHA-3-256 circuits require extensive bit-level constraints. According to zk-SNARK benchmarks, SHA-3-256 for a short input (like 128 bits) costs $\sim 27,000$ con-

straints[8].

Signature verification. Recall that the relation we are interested in is

$$a \cdot s_1 + s_2 + b \cdot r_1 + r_2 + q_F \delta = \sum_{j=1}^{128} c_j \cdot k_j + \mathsf{H}(u) \,,$$

where $a, b, c_1, \ldots, c_{128}$ are public polynomials and $s_1, s_2, r_1, r_2, d, k_1, \ldots, k_{128}, u$ are secret witness polynomials. Now, for each multiplication of public and secret polynomials, we need one R1CS constraint per coefficient. Thus, for $a \cdot s_1$, s_2, $b \cdot r_1$, r_2, and $q_F \cdot d$, each contributes 512 constraints, resulting in a total of 2,560 constraints for the left-hand side. On the right-hand side, each term $c_j \cdot k_j$ requires 512 constraints for the multiplication of their coefficients, resulting in 65536 constraints for the entire sum. Additionally, u, being a secret polynomial, contributes 512 constraints. Together, the right-hand side contributes 66048 constraints. Finally, we need one additional R1CS constraint to enforce the equality between the left-hand side and the right-hand side of the equation. Adding all of these constraints together, we get a total of 68609 R1CS constraints.

PRF computation. To compute the R1CS constraints for the given check, we will break down the individual operations. First, we compute $\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r} \\ \mathbf{g}^{-1}(\mathfrak{i}) \end{bmatrix}$, where $\mathbf{Y}_0 \in \mathbb{Z}_3^{128 \times (1024+\kappa_N)}$ for $\kappa_N = \lceil \log_2(N) \rceil$ is a public matrix and $\begin{bmatrix} \mathbf{r} \\ \mathbf{g}^{-1}(\mathfrak{i}) \end{bmatrix} \in \mathbb{Z}_2^{1024+\kappa_N}$ is part of the witness. The multiplication requires $(1024 + \kappa_N)$ operations, resulting in as many constraints. Next, the inverse gadget decomposition $\mathbf{G}^{-1}$ maps the resulting vector from $\mathbb{Z}_3^{128} \to \mathbb{Z}_2^{256}$, by expanding each of the 128 entries of the input vector into its (2-bit) binary representation. Thus, each of the 128 elements introduces two binary variables, resulting in a total of 256 constraints for the $\mathbf{G}^{-1}$ operation. We then apply the vectorization operation Vec^{-1} to $\mathsf{Coeff}(k_1, \ldots, k_{128})$. The Vec^{-1} operation transforms a vector in $\mathbb{Z}_2^{65536}$ into a matrix in $\mathbb{Z}_2^{256 \times 256}$. This transformation involves matrix multiplication and the Kronecker product of identity matrices, which essentially reshapes the vector into a matrix. Since the result is a 256×256 matrix, and each element in the matrix corresponds to a constraint in the R1CS, the operation introduces 65536 constraints. The resulting matrix and vector are then multiplied to give a new vector in $\mathbb{Z}_2^{256}$. For each element of the resulting vector, the matrix-vector multiplication involves a dot product of a row of the 256×256 matrix with the 256-dimensional vector. Each of these dot products requires 256 scalar multiplications, followed by a summation. Since we have a total of 256 elements in the resulting vector, the operations for this matrix-vector multiplication contributes to $256 \cdot 256 = 65536$ constraints. Finally, we multiply the result with $\mathbf{Y}_1 \in \mathbb{Z}_3^{82 \times 256}$. The total number of operations for this matrix-vector multiplication is $82 \cdot 256 = 20992$ and as many constraints. Summing all of these contributions, the total number of R1CS constraints is $283392 + 128 \cdot \kappa_N$. For $N \leqslant 2^{20}$, this value is bounded from above by 286000.

PKE computation. The proof of computation of the encryption of $\mathbf{m} \in \mathcal{R}^{133}_{q_{\mathsf{PKE}}, d_F}$ according to [LPS10] requires proving

$$\psi_1 = a_1 \cdot s + e_1 \text{ and } \boldsymbol{\psi}_2 = \mathbf{a}_2 \cdot s + \mathbf{e}_2 + p \cdot \mathbf{m} \,,$$

[8] https://tinyurl.com/eth-research-sha-const

for ciphertexts $(\psi_1, \psi_2) \in \mathcal{R}_{q_{\mathsf{PKE}},d_{\mathsf{F}}} \times \mathcal{R}^{133}_{q_{\mathsf{PKE}},d_{\mathsf{F}}}$, public polynomial $a_1 \in \mathcal{R}_{q_{\mathsf{PKE}},d_{\mathsf{F}}}$ and polynomial vector $\mathbf{a}_2 \in \mathcal{R}^{133}_{q_{\mathsf{PKE}},d_{\mathsf{F}}}$ and witness $(s, e_1, \mathbf{e}_2, \mathbf{m}) \in \mathcal{R}_{q_{\mathsf{PKE}},d_{\mathsf{F}}} \times \mathcal{R}_{q_{\mathsf{PKE}},d_{\mathsf{F}}} \times \mathcal{R}^{133}_{q_{\mathsf{PKE}},d_{\mathsf{F}}} \mathcal{R}^{133}_{q_{\mathsf{PKE}},d_{\mathsf{F}}}$.

To enforce the first relation, we need to compute multiply a public polynomial with a secret polynomial. This requires 512 R1CS constraints, one per coefficient; and the addition of a secret polynomial to this product involves another 512 constraints. For the second relation, we notice that each polynomial-vector multiplication requires 512 constraints for each of the 133 entries in the vector, leading to $133 \cdot 512 = 68096$ constraints, and the scalar multiplication also contributes 68096 constraints. Finally, adding up the three vectors contributes $2 \cdot 68096 = 136192$ constraints. The total number of R1CS constraints required to enforce both equations is therefore $1024 + 4 \cdot 68096 = 273408$.

ℓ_2-norm bound. For any polynomial, computing the square of each coefficient requires 512 multiplication constraints, summing these squares adds 511 constraints, and enforcing that the resulting sum is less than or equal to some (public) bound requires approximately 64 constraints via bit-decomposition and range checking. Therefore, each polynomial contributes roughly $512 + 511 + 64 = 1087$ R1CS constraints, and since we need to perform this check for polynomials s_1, s_2, δ, s and e_1, and polynomial vector $\mathbf{e}_2$, the total number of constraints is $4 \cdot 1087 + 133 \cdot 512 + 511 + 64 = 73019$.

Binary coefficient checks. Checking that the coefficients of polynomials $k_1, k_2, \ldots, k_{128}, r_1$ and r_2 are binary, can be done by ensuring that for each coefficient x, the equation $x \cdot (1 - x) = 0$ holds, which requires one R1CS constraint per coefficient. Since each of the polynomials has 512 coefficients, the total number of R1CS constraints is $512 \cdot 130 = 66560$.

6 Lattice-based NI$\underline{\text{B}}$BS from ISIS_f

A minor drawback of our construction 5 is that it requires proving the computation of the hash function in the NIZK proof. Aside from the computational overhead, this additionally necessitates a somewhat strong assumption, since the proof of unforgeability also requires modeling the hash function as a random oracle. While not unusual in practice (see for example [BLNS23a]), this approach may be unsatisfactory for provable security. In this section, we give an alternate construction that is secure under the recently proposed ISIS_f assumption [BLNS23b] which does not require this stronger assumption concerning the random oracle. We begin by recalling the ISIS_f assumption [BLNS23b]:

Definition 6.1 (ISIS_f [BLNS23b]). *Let $q, n, m, \mathfrak{s}, \mathfrak{B}$ be functions of security parameter λ. Also let $f : \mathcal{D} \to \mathbb{Z}_q^n$ be a function on some input domain $\mathcal{D}$ and for $\mathbf{A} \leftarrow\!\!\$\, \mathbb{Z}_q^{n \times m}$, let $\mathcal{O}_{\mathbf{A},f}$ be an oracle that chooses a random input $\alpha \in \mathcal{D}$ and outputs $(\mathbf{s}, \alpha)$ for $\mathbf{s} \in \mathbb{Z}_q^m$ such that $\mathbf{s} \leftarrow\!\!\$\, \mathbf{A}_{\mathfrak{s}}^{-1}(f(\alpha))$. Given $(\mathbf{A}, f)$ and access to $\mathcal{O}_{\mathbf{A},f}$, the ISIS_f problem asks to find $(\mathbf{s}', \alpha') \in \mathbb{Z}_q^m \times \mathcal{D}$, such that $\mathbf{A} \cdot \mathbf{s}' = f(\alpha')$, $\mathbf{s}' \neq \mathbf{s}$ and $\|\mathbf{s}'\| \leqslant \mathfrak{B}$.*

Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}$, and positive integer q be functions of the security parameter λ satisfying the constraints in (1). By $f : [2^m] \to \mathbb{Z}_q^n$ we denote a linear map characterizing the ISIS_f. We give a NI$\underline{\text{B}}$BS construction that is secure under the ISIS_f assumption.

This construction relies on a lattice trapdoor $\mathcal{T} = (\mathcal{T}.\mathsf{TrapGen}, \mathcal{T}.\mathsf{SamplePre})$, a public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ and NIZK proof systems $\mathsf{NIZK}_1 = (\mathsf{NIZK}_1.\mathsf{Setup}, \mathsf{NIZK}_1.\mathsf{Prove}, \mathsf{NIZK}_1.\mathsf{Verify})$, $\mathsf{NIZK}_2 = (\mathsf{NIZK}_2.\mathsf{Setup}, \mathsf{NIZK}_2.\mathsf{Prove}, \mathsf{NIZK}_2.\mathsf{Verify})$ respectively for relations $\mathscr{R}_1^{\mathsf{NI\underline{B}BS},\mathsf{ISIS}_f}$ and $\mathscr{R}_2^{\mathsf{NI\underline{B}BS},\mathsf{ISIS}_f}$, defined as follows:

Relation $\mathscr{R}_1^{\mathsf{NI\underline{B}BS,ISIS}_f}$

Instance: Each instance $\mathfrak{x}$ is interpreted as polynomial matrices $\mathbf{C}_0$ and $\mathbf{C}_1$, an encryption public key $\mathsf{pk}_{\mathsf{PKE}}$, a polynomial vector $\mathbf{t}$ and a ciphertext ϕ.

Witness: Witness $\mathfrak{w}$ consists of secret polynomial vectors $\mathbf{k}$ and $\mathbf{u}$, and the encryption randomness ν.

Membership: $\mathfrak{w}$ is a valid witness for $\mathfrak{x}$ if the following are satisfied:

- $\phi = \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}\|\mathbf{u}; \nu)$
- $\mathbf{t} = \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$
- $\mathbf{k}, \mathbf{u} \in \mathbb{Z}_2^{n' m_N}$

Relation $\mathscr{R}_2^{\mathsf{NI\underline{B}BS,ISIS}_f}$

Instance: Each instance $\mathfrak{x}$ is interpreted as matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0$ and $\mathbf{Y}_1$, a linear map f, a PKE public key $\mathsf{pk}_{\mathsf{PKE}}$, a message vector μ and ciphertext ψ.

Witness: Witness $\mathfrak{w}$ consists of secret vectors $\mathbf{k}, \mathbf{u}, \mathbf{s}$ and $\mathbf{i}$, integer α and PKE randomness ρ.

Membership: $\mathfrak{w}$ is a valid witness for $\mathfrak{x}$ if the following are satisfied:

- $\mathbf{A} \cdot \mathbf{s} = \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u} + f(\alpha)$
- $\mu = \mathbf{Y}_1 \cdot \left(\mathsf{Vec}^{-1}(\mathbf{k}) \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} f(\alpha) \\ \mathbf{i} \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3$
- $\psi = \mathsf{PKE.Enc}\left(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}\|\mathbf{u}\|\mathbf{s}\|\mathbf{i}\|\alpha \; ; \; \rho\right)$
- $\|\mathbf{s}\| \leqslant \mathfrak{B}, \quad \alpha \in [2^m], \quad \mathbf{k}, \mathbf{u} \in \mathbb{Z}_2^{n' m_N}, \quad \mathbf{i} \in \mathbb{Z}_2^{m_N - m'}$.

6.1 Construction

We are now ready to describe our lattice-based NI$\underline{\mathrm{B}}$BS scheme.

- $\mathsf{Setup}(1^\lambda, N) \to \mathsf{pp}$. The public parameter setup algorithm computes $n, m, n', m', \mathfrak{s}, q$ from the security parameter λ as well as the linear map f. It then sets $m_N = m' + \lceil \log_2(N) \rceil$, and samples matrices $\mathbf{Y}_0 \leftarrow\!\!\$\, \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\!\$\, \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\!\$\, \mathbb{Z}_q^{n \times n' m_N}$. It then runs the key generation algorithm of PKE and generates the corresponding public and secret key pair as

$$(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\, \mathsf{PKE.KeyGen}(1^\lambda) .$$

Next, it generates $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$\, \mathsf{NIZK}_1.\mathsf{Setup}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\$\, \mathsf{NIZK}_2.\mathsf{Setup}(1^\lambda)$ and outputs

$$\mathsf{pp} := \left(1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, f, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1\right).$$

The public parameters are fixed once and for all.

- $\mathsf{KeyGen}_S(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{vk})$. The signer's key generation algorithm is the same as in construction 5.

- $\mathsf{KeyGen}_R(\mathsf{pp}) \to (\mathsf{sk}_R, \mathsf{pk}_R)$. The receiver's key generation algorithm samples the PRF key $\mathbf{K} \leftarrow\!\$\, \mathbb{Z}_2^{n' \times m_N}$ and a random vector $\mathbf{u} \leftarrow\!\$\, \mathbb{Z}_2^{n' m_N}$. It then vectorizes $\mathbf{K}$ as $\mathbf{k} := \mathsf{Vec}(\mathbf{K})$, and computes an Ajtai commitment

$$\mathbf{t} := \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}\,.$$

Next, it generates a ciphertext,

$$\phi \leftarrow \mathsf{PKE.Enc}\left(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u}\ ;\ \nu\right)$$

from uniformly sampled randomness ν and the NIZK proof

$$\tau \leftarrow\!\$\, \mathsf{NIZK}_1.\mathsf{Prove}\left(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x} := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}, \phi), \mathfrak{w} := (\mathbf{k}, \mathbf{u}, \nu)\right),$$

It then sets the receiver's secret key as $\mathsf{sk}_R := (\mathbf{K}, \mathbf{u})$, the public key as $\mathsf{pk}_R := (\mathbf{t}, \phi, \tau)$ and outputs them.

- $\mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_R) \to (\mathsf{psig}, \mathsf{nonce})$. Given the receiver's public key $\mathsf{pk}_R =: (\mathbf{t}, \phi, \tau)$, the signer's pre-signature issuance algorithm first checks if

$$\mathsf{NIZK}_1.\mathsf{Verify}(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x} := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}, \phi), \tau) \stackrel{?}{=} 1\ ,$$

otherwise it aborts and outputs $\perp$. It then samples a random integer $\alpha \leftarrow\!\$\, [2^m]$ and, using the secret signing key $\mathsf{sk} =: \mathbf{T}_{\mathbf{A}}$, it computes

$$\mathbf{s} \leftarrow\!\$\, \mathcal{T}.\mathsf{SamplePre}\left(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathbf{t} + f(\alpha), \mathfrak{s}\right),$$

and outputs the pre-signature, $\mathsf{psig} := \mathbf{s}$ and the nonce, $\mathsf{nonce} := \alpha$.

- $\mathsf{Obtain}(\mathsf{sk}_R, \mathsf{vk}, \mathsf{psig}, \mathsf{nonce}) \to [\![\mu, \sigma]\!]$. The receiver's signature obtainment algorithm first parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, $\mathsf{psig} =: \mathbf{s}$, and $\mathsf{nonce} =: \alpha$, it checks if

$$\mathbf{A} \cdot \mathbf{s} \stackrel{?}{=} \mathbf{C}_0 \cdot \mathsf{Vec}(\mathbf{K}) + \mathbf{C}_1 \cdot \mathbf{u} + f(\alpha)\ \wedge\ \|\mathbf{s}\| \stackrel{?}{\leqslant} \mathfrak{B}\ \wedge\ \alpha \stackrel{?}{\in} [2^m]\,.$$

If any check fails it aborts and outputs $\perp$. Otherwise, for each $\mathsf{i} \in [N]$, it computes the message μ_{i} as

$$\mu_{\mathrm{i}} := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} f(\alpha) \\ \mathbf{g}^{-1}(\mathrm{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,,$$

where, $\mathbf{g}^{-1} : \mathbb{Z}_N \to \mathbb{Z}_2^{\log_2(N)}$ ($\log_2(N) = m_N - m'$) is the binary decomposition gadget. With this, it now generates the corresponding ciphertext as

$$\psi_{\mathrm{i}} \leftarrow \mathsf{PKE.Enc}\left(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u} \| \mathbf{s} \| \mathbf{g}^{-1}(\mathrm{i}) \| \alpha \; ; \; \rho_{\mathrm{i}}\right)$$

from uniformly sampled randomness ρ_{i}. It then generates the NIZK proof

$$\pi_{\mathrm{i}} \leftarrow\!\!\$\, \mathsf{NIZK}_2.\mathsf{Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_{\mathrm{i}} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}, \psi_{\mathrm{i}}), \\ \mathfrak{w}_{\mathrm{i}} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}, \mathbf{g}^{-1}(\mathrm{i}), \alpha, \rho_{\mathrm{i}}\right) \end{array} \right),$$

Finally, it outputs $[\![\mu, \sigma]\!] := \{(\mu_1, \sigma_1 := (\psi_1, \pi_1)), \ldots, (\mu_N, \sigma_N := (\psi_N, \pi_N))\}$.

- $\mathsf{Verify}(\mathsf{vk}, \mu, \sigma) \to 0/1$. The signature verification algorithm parses σ as (ψ, π) and outputs 1 (accept) if

$$\mathsf{NIZK}_2.\mathsf{Verify}\left(\mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}, \psi_{\mathrm{i}}), \pi\right) = 1 \,,$$

and 0 (reject) otherwise.

Security. We now state our main theorem concerning the security of our NIBBS scheme.

Theorem 6.2. *Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}$, and positive integer q be functions of the security parameter λ such that (1) is satisfied, and the ISIS_f and $\mathsf{SIS}_{q,n,2n'm_N,\sqrt{2n'm_N}}$ problems are hard for an appropriately chosen function $f : [2^m] \to \mathbb{Z}_q^n$.*

If NIZK_1 is a NIZK for $\mathscr{R}_1^{\mathsf{NIBBS},\mathsf{ISIS}_f}$, NIZK_2 is a NIZK for $\mathscr{R}_2^{\mathsf{NIBBS},\mathsf{ISIS}_f}$ and PKE is an $\mathsf{IND\text{-}CPA}$ secure encryption scheme then, construction 6 is a secure non-interactive batched blind signature scheme.

Proof of theorem. The correctness, succinctness of communication and reusability of construction 6 can be shown similarly to that of the previous construction. So only we prove the following lemmas concerning the one-more unforgeability, and inter- and intra-batch of the protocol in Appendix B.

Lemma 6.3 (One-more unforgeability)**.** *Suppose the ISIS_f and $\mathsf{SIS}_{q,n,2n'm_N,\sqrt{2n'm_N}}$ problems are hard and NIZK_1 and NIZK_2 are sound. Then, construction 6 is one-more unforgeable.*

Lemma 6.4 (Inter-batch blindness)**.** *Suppose the $\mathsf{F}_{\mathbf{K}}$ is a pseudorandom function, NIZK_1 and NIZK_2 are multi-theorem zero-knowledge proof systems and PKE is $\mathsf{IND\text{-}CPA}$ secure. Then, construction 6 is batch inter-batch blind.*

Lemma 6.5 (Intra-batch blindness)**.** *Suppose the $\mathsf{F}_{\mathbf{K}}$ is a pseudorandom function, NIZK_1 and NIZK_2 are multi-theorem zero-knowledge proof systems and PKE is $\mathsf{IND\text{-}CPA}$ secure. Then, construction 6 is intra-batch blind.*

References

[AAB+24] Marius A. Aardal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. Aggregating falcon signatures with LaBRADOR. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 71–106, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[Abe01] Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151, Innsbruck, Austria, May 6–10, 2001. Springer Berlin Heidelberg, Germany.

[ADDG24] Martin R. Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto dark matter on the torus - oblivious PRFs from shallow PRFs and TFHE. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 447–476, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.

[ADDS21] Martin R. Albrecht, Alex Davidson, Amit Deo, and Nigel P. Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 261–289, Virtual Event, May 10–13, 2021. Springer, Cham, Switzerland.

[AG24] Martin R. Albrecht and Kamil Doruk Gür. Verifiable oblivious pseudorandom functions from lattices: Practical-ish and thresholdisable. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 205–237, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219, Hanoi, Vietnam, December 4–8, 2016. Springer Berlin Heidelberg, Germany.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[AKSY22] Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 39–53, Los Angeles, CA, USA, November 7–11, 2022. ACM Press.

[BBKK18] Boaz Barak, Zvika Brakerski, Ilan Komargodski, and Pravesh K. Kothari. Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 649–679, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland.

[BCC04] Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004*, pages 132–145, Washington, DC, USA, October 25–29, 2004. ACM Press.

[BCGY24] Foteini Baldimtsi, Jiaqi Cheng, Rishab Goyal, and Aayush Yadav. Non-interactive blind signatures: Post-quantum and stronger security. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 70–104, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore.

[BEP+21] Pauline Bert, Gautier Eberhart, Lucas Prabel, Adeline Roux-Langlois, and Mohamed Sabt. Implementation of lattice trapdoors on modules and applications. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*, pages 195–214, Daejeon, South Korea, July 20–22, 2021. Springer, Cham, Switzerland.

[BFPV13] Olivier Blazy, Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Short blind signatures. *J. Comput. Secur.*, 21(5):627–661, September 2013.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 699–729, Panaji, India, November 11–14, 2018. Springer, Cham, Switzerland.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 1087–1098, Berlin, Germany, November 4–8, 2013. ACM Press.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany.

[BLNS23a] Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 16–29, Copenhagen, Denmark, November 26–30, 2023. ACM Press.

[BLNS23b] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. A framework for practical anonymous credentials from lattices. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 384–417, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46, Miami, FL, USA, January 6–8, 2003. Springer Berlin Heidelberg, Germany.

[BP14] Abhishek Banerjee and Chris Peikert. New and improved key-homomorphic pseudorandom functions. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 353–370, Santa Barbara, CA, USA, August 17–21, 2014. Springer Berlin Heidelberg, Germany.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1994. Springer Berlin Heidelberg, Germany.

[BS23] Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 518–548, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[CAHL+22] Rutchathon Chairattana-Apirom, Lucjan Hanzlik, Julian Loss, Anna Lysyanskaya, and Benedikt Wagner. PI-cut-choo and friends: Compact blind signatures via parallel instance cut-and-choose and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 3–31, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[CATZ24] Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Pairing-free blind signatures from CDH assumptions. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 174–209, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[CCKK22] Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim. Adventures in crypto dark matter: attacks, fixes and analysis for weak pseudorandom functions. *DCC*, 90(8):1735–1760, 2022.

[CGT06] Sébastien Canard, Matthieu Gaud, and Jacques Traoré. Defeating malicious servers in a blind signatures based voting system. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006*, volume 4107 of *LNCS*, pages 148–153, Anguilla, British West Indies, February 27 – March 2, 2006. Springer Berlin Heidelberg, Germany.

[Cha83] David Chaum. Blind signature system. In David Chaum, editor, *CRYPTO'83*, page 153, Santa Barbara, CA, USA, 1983. Plenum Press, New York, USA.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 517–547, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[DGS+18] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *PoPETs*, 2018(3):164–180, July 2018.

[dPK22] Rafaël del Pino and Shuichi Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Yevgeniy

Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 306–336, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540, Interlaken, Switzerland, May 2–6, 2004. Springer Berlin Heidelberg, Germany.

[EKS+21] Muhammed F. Esgin, Veronika Kuchta, Amin Sakzad, Ron Steinfeld, Zhenfei Zhang, Shifeng Sun, and Shumo Chu. Practical post-quantum few-time verifiable random function with applications to algorand. In Nikita Borisov and Claudia Díaz, editors, *FC 2021, Part II*, volume 12675 of *LNCS*, pages 560–578, Virtual Event, March 1–5, 2021. Springer Berlin Heidelberg, Germany.

[ESLR23] Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and VRFs. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 484–517, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[FHS15] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany.

[FHS19] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, April 2019.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77, Santa Barbara, CA, USA, August 20–24, 2006. Springer Berlin Heidelberg, Germany.

[GG14] Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 477–495, Copenhagen, Denmark, May 11–15, 2014. Springer Berlin Heidelberg, Germany.

[Gha17] Essam Ghadafi. Efficient round-optimal blind signatures in the standard model. In Aggelos Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 455–473, Sliema, Malta, April 3–7, 2017. Springer, Cham, Switzerland.

[GKK+25] Anindya Ganguly, Angshuman Karmakar, Suparna Kundu, Debranjan Pal, and Sumanta Sarkar. Revisiting lattice-based non-interactive blind signature. Cryptology ePrint Archive, Paper 2025/1848, 2025.

[Goo25] Google. Private state tokens. https://developers.google.com/privacy-sandbox/protections/private-state-tokens, 2025. Accessed: 2025-05-2.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[GRS+11] Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648, Santa Barbara, CA, USA, August 14–18, 2011. Springer Berlin Heidelberg, Germany.

[HAB+17] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. TumbleBit: An untrusted bitcoin-compatible anonymous payment hub. In *NDSS 2017*, San Diego, CA, USA, February 26 – March 1, 2017. The Internet Society.

[Han23] Lucjan Hanzlik. Non-interactive blind signatures for random messages. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 722–752, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[HBG16] Ethan Heilman, Foteini Baldimtsi, and Sharon Goldberg. Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In Jeremy Clark, Sarah Meiklejohn, Peter Y. A. Ryan, Dan S. Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, volume 9604 of *Lecture Notes in Computer Science*, pages 43–60. Springer, 2016.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HKL19] Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland.

[HLW23] Lucjan Hanzlik, Julian Loss, and Benedikt Wagner. Rai-choo! Evolving blind signatures to the next level. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 753–783, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[HPZ25] Lucjan Hanzlik, Eugenio Paracucchi, and Riccardo Zanotto. Non-interactive blind signatures from rsa assumption and more. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, pages 365–394, Cham, 2025. Springer Nature Switzerland.

[HS14] Christian Hanser and Daniel Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 491–511,

Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer Berlin Heidelberg, Germany.

[IET25a] IETF. Batched token issuance protocol. https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-batched-tokens-04, 2025. Accessed: 2025-05-2.

[IET25b] IETF. Privacy pass (privacypass). https://datatracker.ietf.org/wg/privacypass/about/, 2025. Accessed: 2025-05-2.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski, Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164, Santa Barbara, CA, USA, August 17–21, 1997. Springer Berlin Heidelberg, Germany.

[JS24] Corentin Jeudy and Olivier Sanders. Improved lattice blind signatures from recycled entropy. Cryptology ePrint Archive, Report 2024/1289, 2024.

[KLR21] Jonathan Katz, Julian Loss, and Michael Rosenberg. Boosting the security of blind signature schemes. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021*, *Part IV*, volume 13093 of *LNCS*, pages 468–492, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[KLX22] Julia Kastner, Julian Loss, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022*, *Part II*, volume 13178 of *LNCS*, pages 468–497, Virtual Event, March 8–11, 2022. Springer, Cham, Switzerland.

[LM18] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. *Journal of Cryptology*, 31(3):774–797, July 2018.

[LNP22a] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Efficient lattice-based blind signatures via gaussian one-time signatures. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022*, *Part II*, volume 13178 of *LNCS*, pages 498–527, Virtual Event, March 8–11, 2022. Springer, Cham, Switzerland.

[LNP22b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022*, *Part II*, volume 13508 of *LNCS*, pages 71–101, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[LNPS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021*, *Part IV*, volume 13093 of *LNCS*, pages 218–248, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[LPS10] Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-key cryptographic primitives provably as secure as subset sum. In Daniele Micciancio, editor, *TCC 2010*,

volume 5978 of *LNCS*, pages 382–400, Zurich, Switzerland, February 9–11, 2010. Springer Berlin Heidelberg, Germany.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

[MRFH25] Thibault Meunier, Cefan Daniel Rubin, and Armando Faz-Hernández. Privacy pass: upgrading to the latest protocol version. https://blog.cloudflare.com/privacy-pass-standard, 2025. Accessed: 2025-02-2.

[PFH+22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[PS96] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 252–265, Kyongju, Korea, November 3–7, 1996. Springer Berlin Heidelberg, Germany.

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

[PZ13] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1.1 (revision 3). Technical report, Microsoft Corporation, December 2013.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[SAB+22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[TZ22] Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 782–811, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland.

[ZCH25] Haoqi Zhang, Xinjian Chen, and Qiong Huang. Lattice-based non-interactive blind signature schemes in the random oracle model. In Joseph K. Liu, Liqun Chen, Shi-Feng Sun, and Xiaoning Liu, editors, *Provable and Practical Security*, pages 289–308, Singapore, 2025. Springer Nature Singapore.

A Security Proofs for Section 5

Theorem 5.1 *Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}, \mathfrak{B}_N = (\mathfrak{s} + n'm_N)\sqrt{m}$, and positive integer q be functions of the security parameter λ satisfying (1) such that the randomized one-more ISIS instance $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ is hard.*

If $\mathsf{F}_{\mathbf{K}}$ is a pseudorandom function, NIZK is a NIZK for $\mathscr{R}^{\mathsf{NI}\underline{\mathsf{B}}\mathsf{BS}}$ and PKE is an IND-CPA secure encryption scheme then, construction 5 is a secure non-interactive batched blind signature scheme.

Proof of theorem. We provide the missing details to conclude the proof of the theorem.

Lemma 5.3 (One-more unforgeability). *Suppose the $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ problem is hard and NIZK is sound. Then, construction 5 is one-more unforgeable.*

Proof. Consider the following types of adversaries:

Type 1. The first type of adversary $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\left\{\mu_j, \sigma_j\right\}_{j\in[NQ+1]}$ such that there exists some $k \in [NQ+1]$ for which that $\mathsf{NIZK.Verify}(\mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_k, \pi_k) = 1$ but there *does not exist* a corresponding witness $\mathfrak{w}_j$ such that $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}$.

Type 2. The second type of adversary $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\left\{\mu_j, \sigma_j\right\}_{i\in[NQ+1]}$ such that for every $k \in [NQ+1]$ there *exists* a witness $\mathfrak{w}_k$ for which $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}$.

Proposition A.1. *Assuming NIZK is sound, any Type 1 adversary has negligible advantage.*

Proof. Assume that such an adversary $\mathcal{A}$ has non-negligible advantage $\eta(\lambda)$. We build a reduction algorithm $\mathcal{B}$ that breaks the NIZK soundness, also with non-negligible advantage as follows:

- The soundness challenger starts by generating $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$ and sending it to $\mathcal{B}$ who then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n\times n'm_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and outputs it.
- Next, $\mathcal{B}$ generates $(\mathsf{sk}, \mathsf{vk}) \leftarrow\!\$\ \mathsf{KeyGen}_S(\mathsf{pp})$ and outputs vk.
- On $\mathcal{A}$'s j^{th} pre-signature query $O_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}})$, $\mathcal{B}$ answers with $(\mathbf{s}^{(j)}, \mathbf{r}^{(j)}) \leftarrow \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(j)}}; \mathbf{r}^{(j)})$ for some $\mathbf{r}^{(j)} \leftarrow\!\$\ \mathbb{Z}_2^m$.
- Finally, when $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\{\mu_j, \sigma_j =: (\psi_j, \pi_j)\}_{j\in[NQ+1]}$, $\mathcal{B}$ sets each and outputs $(\mathfrak{x}_j, \pi_j)$, for $j \leftarrow\!\$\ [NQ+1]$ and $\mathfrak{x}_j = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_j, \psi_j)$.

The existence of $\mathcal{A}$ implies that with probability $1/(NQ+1)$, $\mathsf{NIZKVerify}(\mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_k, \pi_k) = 1$ and there does not exist a $\mathfrak{w}_k$ such that $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}$. Thus, if $\mathcal{A}$ has advantage $\eta(\lambda)$, then $\mathcal{B}$ has advantage $\frac{1}{NQ+1} \cdot \eta(\lambda)$, which is non-negligible. It follows that an adversary of the first must have negligible advantage. □

Proposition A.2. *Assuming the $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ problem is hard, any Type 2 adversary has negligible advantage.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBS one-more unforgeability experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\$ \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\$ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\$ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\$ \mathsf{NIZK.Setup}(1^\lambda)$, and sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$. Next, it samples $(\mathbf{T}_\mathbf{A}, \mathbf{A}) \leftarrow\$ \mathcal{T}.\mathsf{TrapGen}(1^\lambda, n, m, q)$ and $\mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n \times m}$. It sets $\mathsf{sk} := \mathbf{T}_\mathbf{A}$, $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and then sends it and vk to the adversary.
2. On $\mathcal{A}$'s j^{th} pre-signature query $O_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}})$, $\mathcal{B}$ answers with $(\mathbf{s}^{(j)}, \mathbf{r}^{(j)}) \leftarrow \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(j)}}; \mathbf{r}^{(j)})$ for some $\mathbf{r}^{(j)} \leftarrow\$ \mathbb{Z}_2^m$. The adversary repeats this step $Q = \mathsf{poly}(\lambda)$ times.
3. The adversary outputs $(NQ+1)$ message-signature pairs $\left\{\mu_j, \sigma_j\right\}_{j \in [NQ+1]}$ and wins if $1 \leqslant j < k \leqslant (NQ+1)$, $\mu_j \neq \mu_k$ and $\mathsf{Verify}(\mathsf{vk}, \mu_j, \sigma_j) = 1$ for all $j \in [NQ+1]$.

Hybrid 1. This is the same as Hybrid 1, except that the challenger withholds $\mathsf{sk}_{\mathsf{PKE}}$, and on receiving the final message-signature pairs $\{\mu_j, \sigma_j =: (\psi_j, \pi_j)\}_j$ from the adversary, the challenger decrypts ψ_j to extract $(\mathbf{k}_j \| \mathbf{u}_j \| \mathbf{s}_j \| \mathbf{r}_j \| \mathbf{i}_j)$. Clearly this affects no change to the adversary's point of view.

Hybrid 2. This is the same as Hybrid 1, except instead of sampling $\mathbf{C}$ uniformly, it instead samples $\mathbf{R} \leftarrow\$ \mathbb{Z}_2^{m \times n' m_N}$ and sets $\mathbf{C} := \mathbf{A} \cdot \mathbf{R}$.

Since $\mathbf{R}$ is sampled uniformly from $\mathbb{Z}_2^{m \times n' m_N}$, we have $\mathbf{H}_\infty(\mathbf{R}) = mn' m_N > nn' m_N \log q + \Theta(\lambda)$. It follows from the leftover hash lemma (Corollary 3.2) that the following distributions are statistically indistinguishable:

$$\left\{(\mathbf{A}', \mathbf{A}' \cdot \mathbf{R}) \;:\; \mathbf{A}' \leftarrow\$ \mathbb{Z}_q^{n \times m}, \mathbf{R} \leftarrow\$ \mathbb{Z}_2^{m \times n' m_N}\right\} \approx_{\mathsf{s}} \left\{(\mathbf{A}', \mathbf{C}) \;:\; \mathbf{A}' \leftarrow\$ \mathbb{Z}_q^{n \times m}, \mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times n' m_N}\right\}.$$

Although $\mathbf{A}'$ is sampled from $\mathbb{Z}_q^{n \times m}$, the well-distributedness (cf. §3.1) of trapdoor generation implies that the statistical distance between $\mathbf{A}'$ and $\mathbf{A}$ is within $2^{-\Omega(\lambda)}$. Thus,

$$\left\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}) \;:\; (\mathbf{T}_\mathbf{A}, \mathbf{A}) \leftarrow\$ \mathcal{T}.\mathsf{TrapGen}(1^\lambda, n, m, q), \mathbf{R} \leftarrow\$ \mathbb{Z}_2^{m \times n' m_N}\right\} \approx_{\mathsf{s}} \left\{(\mathbf{A}, \mathbf{C}) \;:\; \mathbf{A} \leftarrow\$ \mathbb{Z}_q^{n \times m}, \mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times n' m_N}\right\}.$$

So Hybrid 2 is indistinguishable from Hybrid 1 with all but negligible probability.

Now, suppose that such an adversary $\mathcal{A}$ has non-negligible advantage $\eta(\lambda)$, then, we build a reduction algorithm $\mathcal{B}$ that breaks the $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ assumption as follows:

- The rOM-ISIS challenger starts by generating $\mathbf{A}$ and sends it to $\mathcal{B}$, who behaves as in Hybrid 2, sending the pp to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} hash query $\mathbf{u}^{(j)}$, $\mathcal{B}$ simulates the random oracle by making a syndrome query to the rOM-ISIS challenger, and returning the response $\mathbf{h}^{(j)}$ to $\mathcal{A}$.

▹ On $\mathcal{A}$'s j^{th} pre-signature query $O_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}} =: \mathbf{t}^{(j)}$, $\mathcal{B}$ queries the rOM-ISIS challenger with a preimage query for $\mathbf{t}^{(j)}$ and returns the response $(\mathbf{s}^{(j)}, \mathbf{r}^{(j)})$ as the pre-signature and the nonce.

Next, for every $j \in [NQ+1]$, the reduction $\mathcal{B}$ decrypts the ciphertext ψ_j using $\mathsf{sk}_{\mathsf{PKE}}$ as in Hybrid 1 to obtain $(\mathbf{k}_j, \mathbf{u}_j, \mathbf{s}_j, \mathbf{r}_j, \mathbf{i}_j)$. We therefore have two distinct possibilities:

Case 1. For every $j \in [NQ+1]$, $\mathcal{A}$ queried the random oracle for $\mathbf{u}_j$.

Case 2. There exists some $j \in [NQ+1]$, such that $\mathcal{A}$ did not query the random oracle for $\mathbf{u}_j$.

Considering, first, the latter "bad" case, which implies that $\mathcal{A}$ must never have queried the random oracle on $\mathbf{u}_j$ for some $j \in [NQ+1]$. However, the random oracle model dictates that $\mathsf{H}(\mathbf{u}_j)$ must be undefined, and therefore completely random from the point of view of the adversary. Then, from the previous proposition, since it must hold that $\mathsf{H}(\mathbf{u}_j) = \mathbf{A} \cdot \mathbf{s}_j + \mathbf{B} \cdot \mathbf{r}_j - \mathbf{C} \cdot \mathbf{k}_j$, it follows that the probability of the "bad" event is at most negligible.

Now, in the former "good" case, since for each $j \in [NQ+1]$, $\mathbf{h}_j = \mathsf{H}(\mathbf{u}_j)$, it necessarily holds that each $\mathbf{h}_j$ is in the set $\mathcal{S}$. Moreover, we have that

$$\begin{aligned} \mathbf{h}_j &= \mathbf{A} \cdot \mathbf{s}_j + \mathbf{B} \cdot \mathbf{r}_j - \mathbf{C} \cdot \mathbf{k}_j \\ &= \mathbf{A} \cdot \left(\mathbf{s}_j - \mathbf{R} \cdot \mathbf{k}_i\right) + \mathbf{B} \cdot \mathbf{r}_j \ . \end{aligned}$$

Since $\|\mathbf{s}_i\| \leqslant \mathfrak{B}$ and $\mathbf{k}_i \in \mathbb{Z}_2^{n' m_N}$, and $\mathbf{R}$ is an $m \times n' m_N$ binary matrix,

$$\|\mathbf{s}_i - \mathbf{R} \cdot \mathbf{k}_i\| \leqslant \mathfrak{B} + n' m_N \sqrt{m} = \mathfrak{B}_N \ .$$

Thus, at least one $(\mathbf{s}_i - \mathbf{R} \cdot \mathbf{k}_i)$ is a valid (one-more) $\mathsf{rOM\text{-}ISIS}_{q,n,m,\mathfrak{s},\mathfrak{B}_N}$ solution with probability nearly $\eta(\lambda)$. It follows that the adversary of the second type has the same advantage.

□

Thus, the construction is one-more unforgeable. □

Lemma 5.4 (Inter-batch blindness). *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK *is a multi-theorem zero-knowledge proof system and* PKE *is* IND-CPA *secure. Then, construction 5 is inter-batch blind.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBBS inter-batch blindness experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.

2. For $b \in \{0,1\}$, the challenger samples the PRF key $\mathbf{K}_b \leftarrow\!\$\ \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u}_b \leftarrow\!\$\ \mathbb{Z}_2^\lambda$, and then computes a commitment $\mathbf{t}_b$ to $\mathbf{k}_b := \mathsf{Vec}(\mathbf{K}_b) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t}_b := \mathbf{C} \cdot \mathbf{k}_b + \mathsf{H}(\mathbf{u}_b)$. It then sets $\mathsf{sk}_{R_b} := (\mathbf{K}_b, \mathbf{u}_b)$ and $\mathsf{pk}_{R_b} := \mathbf{t}_b$ and sends $\left(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}\right)$ to the adversary.

3. The adversary is allowed to make a series of $Q = \mathsf{poly}(\lambda)$ queries. In the jth query, it chooses $b^{(j)} \in \{0,1\}$, verification key $\mathsf{vk}^{(j)} := (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, pre-signature $\mathsf{psig}^{(j)} := \mathbf{s}^{(j)}$ and nonce $\mathsf{nonce}^{(j)} := \mathbf{r}^{(j)}$, and sends them as a signature obtain query to $O_{\mathsf{pk}_{R_0},\mathsf{pk}_{R_1}}$. The challenger answers the jth query as follows:

   3.1 The challenger parses $\mathsf{sk}_{R_{b^{(j)}}}$ as $(\mathbf{K}_{b^{(j)}}, \mathbf{u}_{b^{(j)}})$ and then, for $\mathsf{vk} =: (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, it checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k}_{b^{(j)}} + \mathsf{H}(\mathbf{u}_{b^{(j)}})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$.

   3.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{\mathsf{i}}^{(j)} := \mathbf{Y}_1 \cdot \left(\mathbf{K}_{b^{(j)}} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r}^{(j)} \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{\mathsf{i}}^{(j)} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_{b^{(j)}} \| \mathbf{u}_{b^{(j)}} \| \mathbf{s}^{(j)} \| \mathbf{r}^{(j)} \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{\mathsf{i}}^{(j)} \ \right)$$

   from uniformly sampled randomness $\rho_{\mathsf{i}}^{(j)}$. Then it generates the NIZK proof:

$$\pi_{\mathsf{i}}^{(j)} \leftarrow\!\$\ \mathsf{NIZK.Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_{\mathsf{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathsf{i}}^{(j)}, \psi_{\mathsf{i}}^{(j)} \right), \\ \mathfrak{w}_{\mathsf{i}}^{(j)} := \left(\mathbf{k}_{b^{(j)}}, \mathbf{u}_{b^{(j)}}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathsf{i}), \rho_{\mathsf{i}}^{(j)} \right) \end{array} \right),$$

   3.3 Finally, it outputs $\left[\!\left[\mu^{(j)}, \sigma^{(j)} \right]\!\right] := \{(\mu_1^{(j)}, \sigma_1^{(j)} := (\psi_1^{(j)}, \pi_1^{(j)})), \ldots, (\mu_N^{(j)}, \sigma_N^{(j)} := (\psi_N^{(j)}, \pi_N^{(j)}))\}$.

4. The adversary chooses and outputs $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and $(\mathsf{psig}_b := \mathbf{s}_b, \mathsf{nonce}_b := \mathbf{r}_b)$ for each $b \in \{0,1\}$. Then, for each $b \in \{0,1\}$, the challenger does the following:

   4.1 The challenger sets $\mathbf{s}_b =: \mathsf{psig}_b$ and $\mathbf{r}_b =: \mathsf{nonce}_b$ and then parses sk_{R_b} as $(\mathbf{K}_b, \mathbf{u}_b)$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, it checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k}_b + \mathsf{H}(\mathbf{u}_b)$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$.

   4.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{b,\mathsf{i}} := \mathbf{Y}_1 \cdot \left(\mathbf{K}_b \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r}_b \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{b,\mathsf{i}} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_b \| \mathbf{u}_b \| \mathbf{s}_b \| \mathbf{r}_b \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{b,\mathsf{i}} \ \right)$$

   from uniformly sampled randomness $\rho_{b,\mathsf{i}}$. Then it generates the NIZK proof:

$$\pi_{b,\mathsf{i}} \leftarrow\!\$\ \mathsf{NIZK.Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_{b,\mathsf{i}} := \left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathsf{i}}, \psi_{b,\mathsf{i}} \right), \\ \mathfrak{w}_{b,\mathsf{i}} := \left(\mathbf{k}_b, \mathbf{u}_b, \mathbf{s}_b, \mathbf{r}_b, \mathbf{g}^{-1}(\mathsf{i}), \rho_{b,\mathsf{i}} \right) \end{array} \right),$$

   4.3 Finally, it outputs $\left([\![\mu, \sigma]\!]_{\hat{b}}, [\![\mu, \sigma]\!]_{1-\hat{b}} \right)$ where $[\![\mu, \sigma]\!]_b := ((\mu_{b,1}, \sigma_{b,1} := (\psi_{b,1}, \pi_{b,1})), \ldots, (\mu_{b,N}, \sigma_{b,N} := (\psi_{b,N}, \pi_{b,N})))$ for each $b \in \{0,1\}$.

5. The (admissible) adversary outputs $b^* \in \{0,1\}$ and wins if $b^* = \hat{b}$.

Hybrid 1. This is the same as Hybrid 0 except that instead of honestly generating the NIZK proofs in steps 3 and 4, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow \mathsf{NIZK}.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using NIZK.Prove in answering the signing queries as well as computing the final signatures, it runs $\mathsf{NIZK}.\mathcal{S}$ without the witness.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 0 and 1. We give a reduction algorithm $\mathcal{B}$ that breaks the multi-theorem zero-knowledge property of NIZK with non-negligible advantage as follows:

- The NIZK challenger starts by sampling $b' \leftarrow\!\$\ \{0,1\}$. If $b' = 0$, it samples $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$; otherwise, it samples $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK}.\mathcal{S}(1^\lambda)$ and sends $\mathsf{crs}_{\mathsf{NIZK}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\ \{0,1\}$. It then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n\times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 0), $\mathcal{B}$ computes $(\mathsf{sk}_{R_0}, \mathsf{pk}_{R_1})$ and $(\mathsf{sk}_{R_0}, \mathsf{pk}_{R_1})$, and sends $(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1})$ to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(b^{(j)}, \mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k}_{b^{(j)}} + \mathsf{H}(\mathbf{u}_{b^{(j)}})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{\mathrm{i}}^{(j)}$ and the encryption $\psi_{\mathrm{i}}^{(j)}$ under $\rho_{\mathrm{i}}^{(j)}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 3 (Hybrid 0). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{\mathrm{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}^{(j)}, \psi_{\mathrm{i}}^{(j)}\right)$ and $\mathfrak{w}_{\mathrm{i}}^{(j)} := \left(\mathbf{k}_{b^{(j)}}, \mathbf{u}_{b^{(j)}}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathrm{i}), \rho_{\mathrm{i}}^{(j)}\right)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{\mathrm{i}}^{(j)}$. It sends the resulting $\left[\!\left[\mu^{(j)}, \sigma^{(j)}\right]\!\right]$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1))$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k}_b + \mathsf{H}(\mathbf{u}_b)$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{b,\mathrm{i}}$ and the encryption $\psi_{b,\mathrm{i}}$ under $\rho_{b,\mathrm{i}}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 4 (Hybrid 0). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{b,\mathrm{i}} := \Big(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathrm{i}}, \psi_{b,\mathrm{i}}\Big)$ and $\mathfrak{w}_{b,\mathrm{i}} := \Big(\mathbf{k}_b, \mathbf{u}_b, \mathbf{s}_b, \mathbf{r}_b, \mathbf{g}^{-1}(\mathrm{i}), \rho_{b,\mathrm{i}}\Big)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{b,\mathrm{i}}$. It sends the resulting $\left([\![\mu, \sigma]\!]_{\hat{b}}, [\![\mu, \sigma]\!]_{1-\hat{b}}\right)$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received an honestly generated proof from NIZK challenger; otherwise, it outputs 1.

If challenger uses a NIZK simulator, $\mathcal{B}$ perfectly simulates Hybrid 1 to $\mathcal{A}$; otherwise, it simulates Hybrid 0. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the NIZK zero-knowledge challenger.

Hybrid 2. This is the same as Hybrid 1 except that in step 4, instead of honestly computing the encryptions $\psi_{0,1}, \ldots, \psi_{0,N}$ on the secrets, it computes it as an encryption of $\mathbf{0}$.

We further divide the hybrid into a sequence of N sub-hybrids so that Hybrid 1_t is the same as Hybrid 1, except that for $t \in [N]$, the encryptions $\psi_{0,1}, \psi_{0,2}, \ldots, \psi_{0,t}$ are computed as encryptions of $\mathbf{0}$ and the remaining $\psi_{0,t+1} \ldots, \psi_{0,N}$ are computed as encryptions of the respective secrets for that batch. Note that Hybrid 1_N is just Hybrid 2.

Now, suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 1_{t-1} and 1_t. We give a reduction algorithm $\mathcal{B}$ that breaks the IND-CPA security of PKE with non-negligible advantage as follows:

- The IND-CPA challenger chooses a bit $b' \leftarrow\!\$\, \{0,1\}$ and samples $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.Setup}(1^\lambda)$ and sends $\mathsf{pk}_{\mathsf{PKE}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\, \{0,1\}$ and then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\, \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\, \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\, \mathsf{NIZK}.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 1), $\mathcal{B}$ computes $(\mathsf{sk}_{R_0}, \mathsf{pk}_{R_1})$ and $(\mathsf{sk}_{R_0}, \mathsf{pk}_{R_1})$, and sends $(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1})$ to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(b^{(j)}, \mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k}_{b^{(j)}} + \mathsf{H}(\mathbf{u}_{b^{(j)}})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{\mathrm{i}}^{(j)}$ and the encryption $\psi_{\mathrm{i}}^{(j)}$ under $\rho_{\mathrm{i}}^{(j)}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 3 (Hybrid 1). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{\mathrm{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}^{(j)}, \psi_{\mathrm{i}}^{(j)}\right)$ and $\mathfrak{w}_{\mathrm{i}}^{(j)} := \left(\mathbf{k}_{b^{(j)}}, \mathbf{u}_{b^{(j)}}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathrm{i}), \rho_{\mathrm{i}}^{(j)}\right)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{\mathrm{i}}^{(j)}$. It sends the resulting $\left[\!\left[\mu^{(j)}, \sigma^{(j)}\right]\!\right]$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1))$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k}_b + \mathsf{H}(\mathbf{u}_b)$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{b,\mathrm{i}}$ and the encryption $\psi_{1,\mathrm{i}}$ under $\rho_{1,\mathrm{i}}$ for each $\mathrm{i} \in [N]$ and the encryptions $\psi_{0,1}, \psi_{0,2}, \ldots, \psi_{0,t-1}, \psi_{0,t+1}, \ldots, \psi_{0,N}$ as in step 4 (Hybrid 1_{t-1}). However, instead of computing $\psi_{0,t}$ the same way, it sets $\mathbf{m}_0 := \mathbf{k}_0 \| \mathbf{u}_0 \| \mathbf{s}_0 \| \mathbf{r}_0 \| \mathbf{g}^{-1}(\mathrm{i})$ and $\mathbf{m}_1 := \mathbf{0}$, and sends $(\mathbf{m}_0, \mathbf{m}_1)$ to the challenger who responds with the ciphertext $\psi_{0,t}$ (to $\mathbf{m}_{b'}$). The rest of the protocol proceeds exactly as Hybrid 1_{t-1} so that $\mathcal{B}$ sends the resulting $\left(\llbracket \mu, \sigma \rrbracket_{\hat{b}}, \llbracket \mu, \sigma \rrbracket_{1-\hat{b}}\right)$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received a ciphertext of R_0's secrets from the IND-CPA challenger; otherwise, it outputs 1.

If challenger encrypts $\mathbf{m}_0$ in $\psi_{0,t}$, $\mathcal{B}$ perfectly simulates Hybrid 1_{t-1} to $\mathcal{A}$; otherwise, it simulates Hybrid 1_t. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the IND-CPA challenger. Applying this argument to each $t \in [N]$, we get that Hybrids 1 and 2 are indistinguishable.

Hybrid 3. This is the same as Hybrid 2 except that in step 4, instead of honestly computing the encryptions $\psi_{1,1}, \ldots, \psi_{1,N}$ on the secrets, it computes it as an encryption of $\mathbf{0}$.

The fact that Hybrids 2 and 3 are indistinguishable can be shown exactly as for Hybrids 1 and 2.

Hybrid 4. This is the same as Hybrid 3 except that in step 2, instead computing a commitment $\mathbf{t}_0 := \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_0) + \mathsf{H}(\mathbf{u}_0)$ to $\mathbf{K}_0$, the challenger samples $\mathbf{t}_0 \leftarrow\!\$\, \mathbb{Z}_q^n$.

Since $\mathbf{K}_0 \leftarrow\!\$\, \mathbb{Z}_2^{n' \times m_N}$, we have $\mathbf{H}_\infty(\mathsf{Vec}(\mathbf{K}_0)) \geqslant n' m_N$. As $n' m_N > n \log q + \Theta(\lambda)$, we obtain from the leftover hash lemma (Corollary 3.2) the fact that following distributions are statistically indistinguishable:

$$\left\{(\mathbf{C}, \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_0)) \;:\; \mathbf{C} \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}, \mathbf{K}_0 \leftarrow\!\$\, \mathbb{Z}_2^{n' \times m_N}\right\}$$
$$\approx_{\mathsf{s}} \left\{(\mathbf{C}, \boldsymbol{v}) \;:\; \mathbf{C} \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}, \boldsymbol{v} \leftarrow\!\$\, \mathbb{Z}_q^n\right\}$$

Thus the vector $\mathbf{t}_0 = \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_0) + \mathsf{H}(\mathbf{u}_0)$ is statistically indistinguishable from a random vector in $\mathbb{Z}_q^n$. It follows that Hybrids 3 and 4 are statistically indistinguishable.

Hybrid 5. This is the same as Hybrid 4 except that in step 2, instead computing a commitment $\mathbf{t}_1 := \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_1) + \mathsf{H}(\mathbf{u}_1)$ to $\mathbf{K}_1$, the challenger samples $\mathbf{t}_1 \leftarrow\!\$\, \mathbb{Z}_q^n$.

The fact that Hybrids 4 and 5 are statistically indistinguishable can be shown exactly as it was for Hybrids 2 and 3.

Hybrid 6. This is the same as Hybrid 5 except that instead of computing the messages for receiver R_0 as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

We further divide the hybrid into a sequence of N sub-hybrids so that Hybrid 5_t is the same as Hybrid 5, except that for $t \in [N]$, the first t messages for receiver R_0 are sampled uniformly from $\mathbb{Z}_3^n$ and the remaining $N - t$ messages are computed as before. Note that Hybrid 5_N is just Hybrid 5.

Now, suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 5_{t-1} and 5_t. We give a reduction algorithm $\mathcal{B}$ that breaks the pseudorandomness of F with non-negligible advantage as follows:

- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\, \{0,1\}$ and then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\, \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\, \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\, \mathsf{NIZK}.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- The PRF challenger chooses a bit $b' \leftarrow\!\$\, \{0,1\}$ such that if $b' = 0$, then the function is pseudorandom (and truly random otherwise). It then samples the PRF key $\mathbf{K}_0 \leftarrow\!\$\, \mathbb{Z}_2^{n' \times m_N}$.
- Similarly to step 2 (Hybrid 5), for each $b \in \{0,1\}$, $\mathcal{B}$ computes the commitments $\mathbf{t}_b \leftarrow\!\$\, \mathbb{Z}_q^n$ and sends $\left(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}\right)$ to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(b^{(j)}, \mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k}_{b^{(j)}} + \mathsf{H}(\mathbf{u}_{b^{(j)}})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise if $b^{(j)} = 0$ it queries the PRF challenger for an evaluation of $\begin{bmatrix} \mathbf{r}^{(j)} \\ \mathbf{g}^{-1}(\mathrm{i}) \end{bmatrix}$ for each $\mathrm{i} \in [t]$, and then sets the messages $\mu_1^{(j)}, \ldots, \mu_t^{(j)}$ as the challenger's responses, and computes the rest of the messages as in Hybrid 5_{t-1}; if not, then for each $\mathrm{i} \in [N]$, it computes $\mu_{\mathrm{i}}^{(j)}$ as it did in Hybrid

5_{t-1}. For each $\mathsf{i} \in [N]$, it then computes $\psi_{\mathsf{i}}^{(j)}$ as encryptions of $\mathbf{0}$, simulates the NIZK proofs $\pi_{\mathsf{i}}^{(j)}$ with on $\mathfrak{x}_{\mathsf{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathsf{i}}^{(j)}, \psi_{\mathsf{i}}^{(j)}\right)$ and sends $\left(\left[\!\left[\mu^{(j)}, \sigma^{(j)} := \left(\psi^{(j)}, \pi^{(j)}\right)\right]\!\right]\right)$ to $\mathcal{A}$.

- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1))$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k}_b + \mathsf{H}(\mathbf{u}_b)$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise if $b = 0$ it queries the PRF challenger for an evaluation of $\begin{bmatrix} \mathbf{r}_0 \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix}$ for each $\mathsf{i} \in [t]$, and then sets the messages $\mu_{0,1}, \ldots, \mu_{0,t}$ as the challenger's responses, and computes the rest of the messages as in Hybrid 5_{t-1}; if not, then for each $\mathsf{i} \in [N]$, it computes $\mu_{1,\mathsf{i}}$ as it did in Hybrid 5_{t-1}. For each $\mathsf{i} \in [N]$, it then computes $\psi_{b,\mathsf{i}}$ as encryptions of $\mathbf{0}$, simulates the NIZK proofs $\pi_{b,\mathsf{i}}$ with on $\mathfrak{x}_{b,\mathsf{i}} := \Big(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathsf{i}}, \psi_{b,\mathsf{i}}\Big)$. The rest of the protocol proceeds exactly as Hybrid 5_{t-1} so that $\mathcal{B}$ sends the resulting $\left(\llbracket \mu, \sigma \rrbracket_{\hat{b}}, \llbracket \mu, \sigma \rrbracket_{1-\hat{b}}\right)$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received pseudorandom messages for R_0; otherwise, it outputs 1.

If the challenger's function is truly random, then $\mathcal{B}$ perfectly simulates Hybrid 5_{t-1} to $\mathcal{A}$; otherwise, it simulates Hybrid 5_t. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the PRF challenger. Applying this argument to each $t \in [N]$, we get that Hybrids 5 and 6 are indistinguishable.

Hybrid 7. This is the same as Hybrid 6 except that instead of computing the messages for receiver R_1 as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

The fact that Hybrids 6 and 7 are indistinguishable can be shown exactly as for Hybrids 5 and 6.

Note that any adversary has 0 advantage in Hybrid 7. Thus, the construction is inter-batch blind. □

Lemma 5.5 (Intra-batch blindness). *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK *is multi-theorem zero-knowledge proof system and* PKE *is* IND-CPA *secure. Then, construction 5 is intra-batch blind.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBS intra-batch blindness experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
2. The challenger first chooses $\hat{b} \leftarrow\!\$\ \{0,1\}$. It then samples the PRF key $\mathbf{K} \leftarrow\!\$\ \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u} \leftarrow\!\$\ \mathbb{Z}_2^\lambda$, and then computes a commitment $\mathbf{t}$ to $\mathbf{k} := \mathsf{Vec}(\mathbf{K}) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t} := \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$. It then sets $\mathsf{sk}_R := (\mathbf{K}, \mathbf{u})$ and $\mathsf{pk}_R := \mathbf{t}$ and sends pk_R to the adversary.

3. The adversary is allowed to make a series of $Q = \mathsf{poly}(\lambda)$ queries. In the jth query, it chooses verification key $\mathsf{vk}^{(j)} := (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, pre-signature $\mathsf{psig}^{(j)} := \mathbf{s}^{(j)}$ and nonce $\mathsf{nonce}^{(j)} := \mathbf{r}^{(j)}$, and sends them as a signature obtain query to $\mathcal{O}_{\mathsf{pk}_R}$. The challenger answers the jth query as follows:

   3.1 The challenger parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk}^{(j)} =: (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, it checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$.

   3.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{\mathsf{i}}^{(j)} := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r}^{(j)} \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{\mathsf{i}}^{(j)} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u} \| \mathbf{s}^{(j)} \| \mathbf{r}^{(j)} \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{\mathsf{i}}^{(j)} \ \right)$$

   from uniformly sampled randomness $\rho_{\mathsf{i}}^{(j)}$. Then it generates the NIZK proof:

$$\pi_{\mathsf{i}}^{(j)} \leftarrow\!\$\ \mathsf{NIZK.Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_{\mathsf{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathsf{i}}^{(j)}, \psi_{\mathsf{i}}^{(j)} \right), \\ \mathfrak{w}_{\mathsf{i}}^{(j)} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathsf{i}), \rho_{\mathsf{i}}^{(j)} \right) \end{array} \right),$$

   3.3 Finally, it outputs $\left[\!\left[\mu^{(j)}, \sigma^{(j)} \right]\!\right] := ((\mu_1^{(j)}, \sigma_1^{(j)} := (\psi_1^{(j)}, \pi_1^{(j)})), \ldots, (\mu_N^{(j)}, \sigma_N^{(j)} := (\psi_N^{(j)}, \pi_N^{(j)})))$.

4. The adversary chooses and outputs $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and $(\mathsf{psig}_b := \mathbf{s}_b, \mathsf{nonce}_b := \mathbf{r}_b)$ for each $b \in \{0,1\}$, as well as a permutation function $\varphi : [N] \times \{0,1\} \to [N] \times \{0,1\}$ for each $b \in \{0,1\}$. Then, for each $b \in \{0,1\}$, the challenger does the following:

   4.1 The challenger sets $\mathbf{s}_b =: \mathsf{psig}_b$ and $\mathbf{r}_b =: \mathsf{nonce}_b$ and then parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, it checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$.

   4.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{b,\mathsf{i}} := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \mathbf{r} \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{b,\mathsf{i}} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u} \| \mathbf{s}_b \| \mathbf{r}_b \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{b,\mathsf{i}} \ \right)$$

   from uniformly sampled randomness $\rho_{b,\mathsf{i}}$. Then it generates the NIZK proof:

$$\pi_{b,\mathsf{i}} \leftarrow\!\$\ \mathsf{NIZK.Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}}, \mathfrak{x}_{b,\mathsf{i}} := (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathsf{i}}, \psi_{b,\mathsf{i}}), \\ \mathfrak{w}_{b,\mathsf{i}} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}_b, \mathbf{r}_b, \mathbf{g}^{-1}(\mathsf{i}), \rho_{b,\mathsf{i}} \right) \end{array} \right),$$

   4.3 Finally, if $\hat{b} = 0$, it outputs the sequence $\{(\mu_{\mathsf{i},b}, \sigma_{\mathsf{i},b})\}_{\substack{\mathsf{i} \in [N] \\ b \in \{0,1\}}}$. Otherwise, it outputs $\left\{\left(\mu_{\varphi(\mathsf{i},b)}, \sigma_{\varphi(\mathsf{i},b)}\right)\right\}_{\substack{\mathsf{i} \in [N] \\ b \in \{0,1\}}}$.

5. The (admissible) adversary outputs $b^* \in \{0,1\}$ and wins if $b^* = \hat{b}$.

Hybrid 1. This is the same as Hybrid 0 except that instead of honestly generating the NIZK proofs in steps 3 and 4, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow \mathsf{NIZK}.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using NIZK.Prove in answering the signing queries as well as computing the final signatures, it runs $\mathsf{NIZK}.\mathcal{S}$ without the witness.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 0 and 1. We give a reduction algorithm $\mathcal{B}$ that breaks the multi-theorem zero-knowledge property of NIZK with non-negligible advantage as follows:

- The NIZK challenger starts by sampling $b' \leftarrow\!\$\ \{0,1\}$. If $b' = 0$, it samples $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK.Setup}(1^\lambda)$; otherwise, it samples $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\ \mathsf{NIZK}.\mathcal{S}(1^\lambda)$ and sends $\mathsf{crs}_{\mathsf{NIZK}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\ \{0,1\}$. It then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n\times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 0), $\mathcal{B}$ computes $(\mathsf{sk}_R, \mathsf{pk}_R)$ and sends pk_R to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(\mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{\mathrm{i}}^{(j)}$ and the encryption $\psi_{\mathrm{i}}^{(j)}$ under $\rho_{\mathrm{i}}^{(j)}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 3 (Hybrid 0). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{\mathrm{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}^{(j)}, \psi_{\mathrm{i}}^{(j)}\right)$ and $\mathfrak{w}_{\mathrm{i}}^{(j)} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathrm{i}), \rho_{\mathrm{i}}^{(j)}\right)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{\mathrm{i}}^{(j)}$. It sends the resulting $\left[\!\left[\mu^{(j)}, \sigma^{(j)}\right]\!\right]$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1), \varphi)$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{b,\mathrm{i}}$ and the encryption $\psi_{b,\mathrm{i}}$ under $\rho_{b,\mathrm{i}}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 4 (Hybrid 0). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{b,\mathrm{i}} := \left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathrm{i}}, \psi_{b,\mathrm{i}}\right)$ and $\mathfrak{w}_{b,\mathrm{i}} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}_b, \mathbf{r}_b, \mathbf{g}^{-1}(\mathrm{i}), \rho_{b,\mathrm{i}}\right)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{b,\mathrm{i}}$. If $\hat{b} = 0$, it sends the sequence $\{(\mu_{\mathrm{i},b}, \sigma_{\mathrm{i},b})\}_{\substack{\mathrm{i}\in[B] \\ b\in\{0,1\}}}$; and otherwise sends $\left\{\left(\mu_{\varphi(\mathrm{i},b)}, \sigma_{\varphi(\mathrm{i},b)}\right)\right\}_{\substack{\mathrm{i}\in[B] \\ b\in\{0,1\}}}$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received an honestly generated proof from NIZK challenger; otherwise, it outputs 1.

If challenger uses a NIZK simulator, $\mathcal{B}$ perfectly simulates Hybrid 1 to $\mathcal{A}$; otherwise, it simulates Hybrid 0. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the NIZK zero-knowledge challenger.

Hybrid 2. This is the same as Hybrid 1 except that in step 4, instead of honestly computing the encryptions $\psi_{0,1}, \ldots, \psi_{0,N}$ on the secrets, it computes it as an encryption of $\mathbf{0}$.

We further divide the hybrid into a sequence of N sub-hybrids so that Hybrid 1_t is the same as Hybrid 1, except that for $t \in [N]$, the encryptions $\psi_{0,1}, \psi_{0,2}, \ldots, \psi_{0,t}$ are computed as encryptions of $\mathbf{0}$ and the remaining $\psi_{0,t+1} \ldots, \psi_{0,N}$ are computed as encryptions of the respective secrets for that batch. Note that Hybrid 1_N is just Hybrid 2.

Now, suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 1_{t-1} and 1_t. We give a reduction algorithm $\mathcal{B}$ that breaks the IND-CPA security of PKE with non-negligible advantage as follows:

- The IND-CPA challenger chooses a bit $b' \leftarrow\!\$\, \{0,1\}$ and samples $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.Setup}(1^\lambda)$ and sends $\mathsf{pk}_{\mathsf{PKE}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\, \{0,1\}$ and then samples matrices $\mathbf{Y}_0 \leftarrow\!\$\, \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\, \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C} \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\$\, \mathsf{NIZK}.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 1), $\mathcal{B}$ computes $(\mathsf{sk}_R, \mathsf{pk}_R)$ and sends pk_R to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(\mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \mathbf{r}^{(j)} = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{\mathrm{i}}^{(j)}$ and the encryption $\psi_{\mathrm{i}}^{(j)}$ under $\rho_{\mathrm{i}}^{(j)}$ for each $\mathrm{i} \in [N]$ using $\mathsf{pk}_{\mathsf{PKE}}$ as in step 3 (Hybrid 1). Then, instead of computing the proof by itself, it queries the NIZK challenger on $\mathfrak{x}_{\mathrm{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}^{(j)}, \psi_{\mathrm{i}}^{(j)}\right)$ and $\mathfrak{w}_{\mathrm{i}}^{(j)} := \left(\mathbf{k}\,\mathbf{u}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{g}^{-1}(\mathrm{i}), \rho_{\mathrm{i}}^{(j)}\right)$ for each $\mathrm{i} \in [N]$, and obtains the corresponding $\pi_{\mathrm{i}}^{(j)}$. It sends the resulting $\left[\!\left[\mu^{(j)}, \sigma^{(j)}\right]\!\right]$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1), \varphi)$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it computes each $\mu_{b,\mathrm{i}}$ and the encryption $\psi_{1,\mathrm{i}}$ under $\rho_{1,\mathrm{i}}$ for each $\mathrm{i} \in [N]$ and the encryptions $\psi_{0,1}, \psi_{0,2}, \ldots, \psi_{0,t-1}, \psi_{0,t+1}, \ldots, \psi_{0,N}$ as in step 4 (Hybrid 1_{t-1}). However, instead of computing $\psi_{0,t}$ the same way, it sets $\mathbf{m}_0 := \mathbf{k} \| \mathbf{u} \| \mathbf{s}_0 \| \mathbf{r}_0 \| \mathbf{g}^{-1}(\mathrm{i})$ and $\mathbf{m}_1 := \mathbf{0}$, and sends $(\mathbf{m}_0, \mathbf{m}_1)$ to the challenger who responds with the ciphertext $\psi_{0,t}$ (to $\mathbf{m}_{b'}$). The rest of the protocol proceeds exactly as Hybrid 1_{t-1} so that $\mathcal{B}$ send the sequence $\{(\mu_{\mathrm{i},b}, \sigma_{\mathrm{i},b})\}_{\substack{\mathrm{i} \in [B] \\ b \in \{0,1\}}}$; and otherwise sends $\left\{\left(\mu_{\varphi(\mathrm{i},b)}, \sigma_{\varphi(\mathrm{i},b)}\right)\right\}_{\substack{\mathrm{i} \in [B] \\ b \in \{0,1\}}}$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received a ciphertext of R_0's secrets from the IND-CPA challenger; otherwise, it outputs 1.

If challenger encrypts $\mathbf{m}_0$ in $\psi_{0,t}$, $\mathcal{B}$ perfectly simulates Hybrid 1_{t-1} to $\mathcal{A}$; otherwise, it simulates Hybrid 1_t. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the IND-CPA challenger. Applying this argument to each $t \in [N]$, we get that Hybrids 1 and 2 are indistinguishable.

Hybrid 3. This is the same as Hybrid 2 except that in step 4, instead of honestly computing the encryptions $\psi_{1,1},\ldots,\psi_{1,N}$ on the secrets, it computes it as an encryption of $\mathbf{0}$.

The fact that Hybrids 2 and 3 are indistinguishable can be shown exactly as for Hybrids 1 and 2.

Hybrid 4. This is the same as Hybrid 3 except that in step 2, instead computing a commitment $\mathbf{t} := \mathbf{C}\cdot\mathsf{Vec}(\mathbf{K}) + \mathsf{H}(\mathbf{u})$ to $\mathbf{K}$, the challenger samples $\mathbf{t}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^n$.

Since $\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n'\times m_N}$, we have $\mathbf{H}_\infty(\mathsf{Vec}(\mathbf{K})) \geqslant n'm_N$. As $n'm_N > n\log q + \Theta(\lambda)$, we obtain from the leftover hash lemma (Corollary 3.2) the fact that following distributions are statistically indistinguishable:

$$\left\{(\mathbf{C}, \mathbf{C}\cdot\mathsf{Vec}(\mathbf{K})) \ :\ \mathbf{C} \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times n'm_N}, \mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n'\times m_N}\right\}$$
$$\approx_{\mathsf{s}} \left\{(\mathbf{C}, \boldsymbol{v}) \ :\ \mathbf{C} \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times n'm_N}, \boldsymbol{v} \leftarrow\!\!\$\ \mathbb{Z}_q^{n}\right\}$$

Thus the vector $\mathbf{t} = \mathbf{C}\cdot\mathsf{Vec}(\mathbf{K}) + \mathsf{H}(\mathbf{u})$ is statistically indistinguishable from a random vector in $\mathbb{Z}_q^n$. It follows that Hybrids 3 and 4 are statistically indistinguishable.

Hybrid 5. This is the same as Hybrid 4 except that instead of computing the messages for receiver R as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

We further divide the hybrid into a sequence of N sub-hybrids so that Hybrid 4_t is the same as Hybrid 4, except that for $t \in [N]$, the first t messages for receiver R are sampled uniformly from $\mathbb{Z}_3^n$ and the remaining $N-t$ messages are computed as before. Note that Hybrid 4_N is just Hybrid 5.

Now, suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 4_{t-1} and 4_t. We give a reduction algorithm $\mathcal{B}$ that breaks the pseudorandomness of F with non-negligible advantage as follows:

- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\!\$\ \{0,1\}$ and then samples matrices $\mathbf{Y}_0 \leftarrow\!\!\$\ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\!\$\ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C} \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times n'm_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{NIZK}.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C})$ and sends it to the adversary.
- The PRF challenger chooses a bit $b' \leftarrow\!\!\$\ \{0,1\}$ such that if $b' = 0$, then the function is pseudorandom (and truly random otherwise). It then samples the PRF key $\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n'\times m_N}$.
- Similarly to step 2 (Hybrid 4), for each $b \in \{0,1\}$, $\mathcal{B}$ computes the commitment $\mathbf{t} \leftarrow\!\!\$\ \mathbb{Z}_q^n$ and sends pk_R to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} signature obtain query $\left(\mathbf{A}^{(j)}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)}\right)$, $\mathcal{B}$ checks if $\mathbf{A}^{(j)}\cdot\mathbf{s}^{(j)} + \mathbf{B}^{(j)}\cdot\mathbf{r}^{(j)} = \mathbf{C}\cdot\mathbf{k} + \mathsf{H}(\mathbf{u})$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\mathbf{r}^{(j)} \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it queries the PRF challenger for an evaluation of $\begin{bmatrix}\mathbf{r}^{(j)} \\ \mathbf{g}^{-1}(\mathfrak{i})\end{bmatrix}$ for each $\mathfrak{i} \in [t]$, and then sets the messages $\mu_1^{(j)},\ldots,\mu_t^{(j)}$ as the challenger's responses, and computes the rest of the messages as in Hybrid 4_{t-1}; if not, then for each $\mathfrak{i} \in [N]$, it computes $\mu_{\mathfrak{i}}^{(j)}$ as it did in Hybrid 4_{t-1}. For each $\mathfrak{i} \in [N]$, it then computes $\psi_{\mathfrak{i}}^{(j)}$ as encryptions of $\mathbf{0}$, simulates the NIZK proofs $\pi_{\mathfrak{i}}^{(j)}$ with on $\mathfrak{x}_{\mathfrak{i}}^{(j)} := \Big(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathfrak{i}}^{(j)}, \psi_{\mathfrak{i}}^{(j)}\Big)$ and sends $\left(\left[\!\left[\mu^{(j)}, \sigma^{(j)} := \left(\psi^{(j)}, \pi^{(j)}\right)\right]\!\right]\right)$ to $\mathcal{A}$.

- When $\mathcal{A}$ sends $(\mathbf{A}, (\mathbf{s}_0, \mathbf{r}_0), (\mathbf{s}_1, \mathbf{r}_1), \varphi)$, for each $b \in \{0,1\}$, $\mathcal{B}$ checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \mathbf{r}_b = \mathbf{C} \cdot \mathbf{k} + \mathsf{H}(\mathbf{u})$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\mathbf{r}_b \in \mathbb{Z}_2^m$. If any check fails it aborts and outputs $\perp$. Otherwise it queries the PRF challenger for an evaluation of $\begin{bmatrix} \mathbf{r}_b \\ \mathbf{g}^{-1}(\mathrm{i}) \end{bmatrix}$ for each $\mathrm{i} \in [t]$, and then sets the messages $\mu_{b,1}, \ldots, \mu_{b,t}$ as the challenger's responses, and computes the rest of the messages as in Hybrid 4_{t-1} for each $b \in \{0,1\}$. For each $\mathrm{i} \in [N]$, it then computes $\psi_{b,\mathrm{i}}$ as encryptions of $\mathbf{0}$, simulates the NIZK proofs $\pi_{b,\mathrm{i}}$ with on $\mathfrak{x}_{b,\mathrm{i}} := \left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Y}_0, \mathbf{Y}_1, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathrm{i}}, \psi_{b,\mathrm{i}}\right)$. The rest of the protocol proceeds exactly as Hybrid 4_{t-1} so that $\mathcal{B}$ send the sequence $\{(\mu_{\mathrm{i},b}, \sigma_{\mathrm{i},b})\}_{\substack{\mathrm{i}\in[N] \\ b\in\{0,1\}}}$; and otherwise sends $\left\{\left(\mu_{\varphi(\mathrm{i},b)}, \sigma_{\varphi(\mathrm{i},b)}\right)\right\}_{\substack{\mathrm{i}\in[N] \\ b\in\{0,1\}}}$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received pseudorandom messages for R_0; otherwise, it outputs 1.

If the challenger's function is truly random, then $\mathcal{B}$ perfectly simulates Hybrid 4_{t-1} to $\mathcal{A}$; otherwise, it simulates Hybrid 4_t. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the PRF challenger. Applying this argument to each $t \in [N]$, we get that Hybrids 4 and 5 are indistinguishable.

Note that any adversary has 0 advantage in Hybrid 5. Thus, the construction is intra-batch blind. □

This completes the proof of the theorem.

B Security Proofs for Section 6

Theorem 6.2 *Let $N \in \mathbb{N}$ be the batch size, and let parameters $n, m, n', m', \mathfrak{s}, \mathfrak{B} = \mathfrak{s}\sqrt{m}$, and positive integer q be functions of the security parameter λ such that (1) is satisfied, and the ISIS_f and $\mathsf{SIS}_{q,n,2n'm_N,\sqrt{2n'm_N}}$ problems are hard for an appropriately chosen function $f : [2^m] \to \mathbb{Z}_q^n$.*

If NIZK_1 is a NIZK for $\mathscr{R}_1^{\mathsf{NI\underline{B}BS,ISIS}_f}$, NIZK_2 is a NIZK for $\mathscr{R}_2^{\mathsf{NI\underline{B}BS,ISIS}_f}$ and PKE is an $\mathsf{IND\text{-}CPA}$ secure encryption scheme. Then, Construction 6 is a secure non-interactive batched blind signature scheme.

Proof of theorem. The correctness, succinctness of communication and reusability of construction 6 can be shown similarly to that of the previous construction. So only we prove the following lemmas concerning the one-more unforgeability, and inter- and intra-batch blindness of the protocol.

Lemma 6.3 (One-more unforgeability). *Suppose the ISIS_f and $\mathsf{SIS}_{q,n,2n'm_N,\sqrt{2n'm_N}}$ problems are hard and NIZK_1 and NIZK_2 are sound. Then, construction 6 is one-more unforgeable.*

Proof. Consider the following types of adversaries:

Type 1. The first type of adversary $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\left\{\mu_j, \sigma_j\right\}_{j\in[NQ+1]}$ such that there exists some $k \in [NQ+1]$ for which that $\mathsf{NIZK}_2.\mathsf{Verify}(\mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_k, \pi_k) = 1$ but there *does not exist* a corresponding witness $\mathfrak{w}_j$ such that $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}_2^{\mathsf{NI\underline{B}BS,ISIS}_f}$.

Type 2. The second type of adversary $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\left\{\mu_j, \sigma_j\right\}_{i\in[NQ+1]}$ such that for every $k \in [NQ+1]$ there *exists* a witness $\mathfrak{w}_k$ for which $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}_2^{\mathsf{NIBBS},\mathsf{ISIS}_f}$.

Proposition B.1. *Assuming* NIZK_2 *is sound, then any Type 1 adversary has negligible advantage.*

Proof. Assume that such an adversary $\mathcal{A}$ has non-negligible advantage $\eta(\lambda)$. We build a reduction algorithm $\mathcal{B}$ that breaks the NIZK_2 soundness, also with non-negligible advantage as follows:

- The soundness challenger starts by generating $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\$ \mathsf{NIZK}_2.\mathsf{Setup}(1^\lambda)$ and sending it to $\mathcal{B}$ who then sets it as $\mathsf{crs}_{\mathsf{NIZK}_2}$ and samples $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\$ \mathsf{NIZK}_1.\mathsf{Setup}(1^\lambda)$, polynomials samples matrices $\mathbf{Y}_0 \leftarrow\$ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\$ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\$ \mathbb{Z}_q^{n\times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\$ \mathsf{PKE}.\mathsf{KeyGen}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, f, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and outputs it.
- Next, $\mathcal{B}$ generates $(\mathsf{sk}, \mathsf{vk}) \leftarrow\$ \mathsf{KeyGen}_S(\mathsf{pp})$ and outputs vk.
- On $\mathcal{A}$'s j^{th} pre-signature query $\mathcal{O}_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}})$, $\mathcal{B}$ answers with $(\mathbf{s}^{(j)}, \alpha^{(j)}) \leftarrow \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(j)}}; \alpha^{(j)})$ for some $\alpha^{(j)} \leftarrow\$ [2^m]$.
- Finally, when $\mathcal{A}$ outputs $(NQ+1)$ valid message and signature pairs $\{\mu_j, \sigma_j =: (\psi_j, \pi_j)\}$, $\mathcal{B}$ sets each $\mathfrak{x}_j := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_i, \psi_i)$ and outputs $(\mathfrak{x}_j, \pi_j)$, for $j \leftarrow\$ [NQ+1]$.

The existence of $\mathcal{A}$ implies that with probability $1/(NQ+1)$, $\mathsf{NIZKVerify}(\mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_k, \pi_k) = 1$ and there does not exist a $\mathfrak{w}_k$ such that $(\mathfrak{x}_k, \mathfrak{w}_k) \in \mathscr{R}$. Thus, if $\mathcal{A}$ has advantage $\eta(\lambda)$, then $\mathcal{B}$ has advantage $\frac{1}{NQ+1} \cdot \eta(\lambda)$, which is non-negligible. It follows that an adversary of the first must have negligible advantage. □

Proposition B.2. *Assuming* ISIS_f, $\mathsf{SIS}_{q,n,2n'm_N,\sqrt{2n'm_N}}$ *problems are hard and* NIZK_1 *is sound, then any Type 2 adversary has negligible advantage.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBS one-more unforgeability experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\$ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\$ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\$ \mathbb{Z}_q^{n\times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\$ \mathsf{PKE}.\mathsf{KeyGen}(1^\lambda)$ and sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, f, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$. Next, it samples $(\mathbf{T}_{\mathbf{A}}, \mathbf{A}) \leftarrow\$ \mathcal{T}.\mathsf{TrapGen}(1^\lambda, n, m, q)$ and $\mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n\times m}$. It sets $\mathsf{sk} := \mathbf{T}_{\mathbf{A}}$, $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and then sends it and vk to the adversary.
2. On $\mathcal{A}$'s j^{th} pre-signature query $\mathcal{O}_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}})$, $\mathcal{B}$ answers with $(\mathbf{s}^{(j)}, \alpha^{(j)}) \leftarrow \mathsf{Issue}(\mathsf{sk}, \mathsf{pk}_{R^{(j)}}; \alpha^{(j)})$ for some $\alpha^{(j)} \leftarrow\$ [2^m]$. The adversary repeats this step $Q = \mathsf{poly}(\lambda)$ times.
3. The adversary outputs $(NQ+1)$ message-signature pairs $\left\{\mu_j, \sigma_j\right\}_{j\in[NQ+1]}$ and wins if $1 \leqslant j < k \leqslant (NQ+1)$, $\mu_j \neq \mu_k$ and $\mathsf{Verify}(\mathsf{vk}, \mu_j, \sigma_j) = 1$ for all $j \in [NQ+1]$.

Hybrid 1. This is the same as Hybrid 1, except that the challenger withholds $\mathsf{sk}_{\mathsf{PKE}}$, and on receiving the final message-signature pairs $\{\mu_j, \sigma_j =: (\psi_j, \pi_j)\}$ from the adversary, the challenger decrypts ψ_j to extract $(\mathbf{k}_j \| \mathbf{u}_j \| \mathbf{s}_j \| \mathbf{i}_j \| \alpha_j)$. Clearly this affects no change to the adversary's point of view.

Now, we divide our analysis into the following cases:

Case 1. The number of distinct α_j for $j \in [NQ+1]$ is greater than Q.

Case 2. The number of distinct α_j for $j \in [NQ+1]$ is exactly Q.

Proposition B.3. *An adversary has negligible advantage in case 1.*

Proof. In the first case, we will use the additional forgery to break the ISIS_f assumption. To that end, we continue our sequence of hybrids as follows:

Hybrid 2. This is the same as Hybrid 1, except instead of sampling $\mathbf{C}_0, \mathbf{C}_0$ uniformly, it instead samples $\mathbf{R}_0 \hookleftarrow \mathbb{Z}_2^{m \times n' m_N}$ and $\mathbf{R}_1 \hookleftarrow \mathbb{Z}_2^{m \times n' m_N}$ and sets $\mathbf{C}_0 := \mathbf{A} \cdot \mathbf{R}_0$ and $\mathbf{C}_1 := \mathbf{A} \cdot \mathbf{R}_1$.

Since $\mathbf{R}_0 \hookleftarrow \mathbb{Z}_2^{m \times n' m_N}$ (the case for $\mathbf{R}_1$ is essentially identical), we have $\mathbf{H}_\infty(\mathbf{R}_0) = mn'm_N > nn'm_N \log q + \Theta(\lambda)$. It follows from the leftover hash lemma (Corollary 3.2) that the following distributions are statistically indistinguishable:

$$\left\{(\mathbf{A}', \mathbf{A}' \cdot \mathbf{R}_0) \;:\; \mathbf{A}' \hookleftarrow \mathbb{Z}_q^{n \times m}, \mathbf{R}_0 \hookleftarrow \mathbb{Z}_2^{m \times n' m_N}\right\} \\ \approx_{\mathsf{s}} \left\{(\mathbf{A}', \mathbf{C}_0) \;:\; \mathbf{A}' \hookleftarrow \mathbb{Z}_q^{n \times m}, \mathbf{C}_0 \hookleftarrow \mathbb{Z}_q^{n \times n' m_N}\right\}.$$

Although $\mathbf{A}'$ is sampled from $\mathbb{Z}_q^{n \times m}$, the well-distributedness (cf. §3.1) of trapdoor generation implies that the statistical distance between $\mathbf{A}'$ and $\mathbf{A}$ is within $2^{-\Omega(\lambda)}$. Thus,

$$\left\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}_0) \;:\; (\mathbf{T}_{\mathbf{A}}, \mathbf{A}) \hookleftarrow \mathcal{T}.\mathsf{TrapGen}(1^\lambda, n, m, q), \mathbf{R}_0 \hookleftarrow \mathbb{Z}_2^{m \times n' m_N}\right\} \\ \approx_{\mathsf{s}} \left\{(\mathbf{A}, \mathbf{C}_0) \;:\; \mathbf{A} \hookleftarrow \mathbb{Z}_q^{n \times m}, \mathbf{C}_0 \hookleftarrow \mathbb{Z}_q^{n \times n' m_N}\right\}.$$

So Hybrid 2 is indistinguishable from Hybrid 1 with all but negligible probability. Now suppose that such an adversary $\mathcal{A}$ has non-negligible advantage $\eta(\lambda)$. As in the previous construction, we build a reduction algorithm $\mathcal{B}$ that breaks the ISIS_f as follows:

- The ISIS_f challenger starts by generating $\mathbf{A}$ and sends it to $\mathcal{B}$, who behaves as in Hybrid 2, sending the pp to $\mathcal{A}$.
- On $\mathcal{A}$'s j^{th} pre-signature query $\mathcal{O}_{\mathsf{sk}}(\mathsf{pk}_{R^{(j)}} =: (\mathbf{t}^{(j)}, \tau^{(j)}, \phi^{(j)})$, $\mathcal{B}$ first uses $\mathsf{sk}_{\mathsf{PKE}}$ to extract $(\mathbf{k}^{(j)} \| \mathbf{u}^{(j)})$ from $\phi^{(j)}$. It then queries the ISIS_f challenger who responds with a tuple $(\mathbf{s}^{(j)}, \alpha^{(j)})$ such that $\mathbf{A} \cdot \mathbf{s}^{(j)} = f(\alpha^{(j)})$. The reduction $\mathcal{B}$ then sets

  $$\mathbf{s}'^{(j)} := \left(\mathbf{s}^{(j)} + \mathbf{R}_0 \cdot \mathbf{k}^{(j)} + \mathbf{R}_1 \cdot \mathbf{u}^{(j)}\right).$$

  As before, $\mathbf{s}'^{(j)}$ is a Gaussian shifted by $\mathbf{R}_0 \cdot \mathbf{k}^{(j)} + \mathbf{R}_1 \cdot \mathbf{u}^{(j)}$, which can be converted into a zero-centered distribution, using the rejection sampling procedure according to the following lemma.

Lemma B.4 ([Lyu12, Lemma 4.6]). *Let $V \subseteq \mathcal{R}^m$ be a set of polynomials with norm at most $\mathfrak{B}$ and $\xi : V \to [0,1]$ be a probability distribution. Fix the standard deviation $\sigma = \gamma\mathfrak{B}$ for $\gamma = O(\sqrt{\lambda})$ and let*

$$M = \exp\left(\frac{1}{\gamma}\sqrt{\frac{2\lambda+1}{\log e}} + \frac{1}{2\gamma^2}\right) = O(1) \ .$$

Then consider an algorithm that samples $\mathbf{v} \leftarrow\!\$\ \xi$, $\mathbf{x} \leftarrow\!\$\ \mathcal{D}_{\mathcal{R}^m,\sigma}$, sets $\mathbf{z} \coloneqq \mathbf{x} + \mathbf{v}$ and accepts with probability $\min\left(\frac{1}{M} \cdot \exp\left(\frac{-2\langle \mathbf{z},\mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\right), 1\right)$, and rejects otherwise. Then, the distribution over $(\mathbf{v},\mathbf{z})$, conditioned on the algorithm accepting, is within statistical distance of $2^{-\lambda}$ of the distribution $\xi \times \mathcal{D}_{\mathcal{R}_m,\sigma}$.

Concretely, $\mathcal{B}$ repeatedly queries the ISIS_f challenger until the algorithm accepts. For an appropriately chosen rejection sampling parameter $\gamma = O(\sqrt{\lambda})$, the algorithm accepts with probability $\left(1 - 2^{-\lambda}\right) \cdot \exp\left(-\frac{12}{\gamma} - \frac{1}{\gamma^2}\right)$.

Once $\mathcal{B}$ finds and accepts such an $\mathbf{s}'^{(j)}$, it sends $(\mathbf{s}'^{(j)}, \alpha^{(j)})$ to $\mathcal{A}$. Since

$$\mathbf{A} \cdot \mathbf{s}'^{(j)} = \mathbf{t}^{(j)} + f(\alpha^{(j)}) \ ,$$

and $\left\|\mathbf{s}'^{(j)}\right\| \leqslant \mathfrak{B}$, this is a valid query response.

Next, for every $j \in [NQ+1]$, the reduction $\mathcal{B}$ decrypts the ciphertext ψ_j using $\mathsf{sk}_{\mathsf{PKE}}$ as in Hybrid 1, and sets $\mathbf{t}_j = \mathbf{C}_0 \cdot \mathbf{k}_j + \mathbf{C}_1 \cdot \mathbf{u}_j$. We therefore have two distinct possibilities:

Case 1.1 For every $j \in [NQ+1]$, $\mathcal{A}$ queried O_{sk} for $(\mathbf{t}_j, \cdot, \cdot)$.

Notice that since $\mathcal{A}$ can make at most Q queries, by a simple pigeonhole argument, there must be some $\mathbf{t}^*$ such that it is extracted from $N+1$ signatures. Let $\left\{\mu_j^*, (\psi_j^*, \pi_j^*)\right\}_{j\in[N+1]}$ denote the message-signature pairs with $\mathbf{t}^*$ as the underlying public key. Then, since for every $1 \leqslant j < k \leqslant N+1$, we necessarily have that $\mathsf{F}_{\mathsf{Vec}^{-1}(\mathbf{k}_j^*)}(\alpha_j^*, \mathsf{i}_j^*) = \mu_j^* \neq \mu_k^* = \mathsf{F}_{\mathsf{Vec}^{-1}(\mathbf{k}_k^*)}(\alpha_k^*, \mathsf{i}_k^*)$. However, recall that by assumption we have for all $k' \in [NQ+1]$

$$\left(\begin{array}{c} \mathfrak{x}_{k'} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{k'}, \psi_{k'}), \\ \mathfrak{w}_{k'} \coloneqq \left(\mathbf{k}_{k'}, \mathbf{u}_{k'}, \mathbf{s}_{k'}, \mathbf{g}^{-1}(\mathsf{i}_{k'}), \alpha_{k'}, \rho_{k'}\right) \end{array}\right) \in \mathscr{R}_2^{\mathsf{NIBBS},\mathsf{ISIS}_f} \ ,$$

which in particular implies that $1 \leqslant \mathsf{i}_j^*, \mathsf{i}_k^* \leqslant N$, so that there exists some $j \neq k$ such that $\mathsf{i}_j^* = \mathsf{i}_k^* = \mathsf{i}^*$ and $\mathsf{F}_{\mathsf{Vec}^{-1}(\mathbf{k}_j^*)}(\alpha_j^*, \mathsf{i}^*) \neq \mathsf{F}_{\mathsf{Vec}^{-1}(\mathbf{k}_k^*)}(\alpha_k^*, \mathsf{i}^*)$. Thus, either $\mathsf{Vec}^{-1}(\mathbf{k}_j^*) \neq \mathsf{Vec}^{-1}(\mathbf{k}_k^*)$, or $\alpha_j^* \neq \alpha_k^*$.

Now, if $\mathcal{A}$ received, both, α_j^* and α_k^* from $\mathcal{B}$ (who in turn received it from the ISIS_f challenger) and $\alpha_j^* = \alpha_k^* = \alpha^*$, then certainly $\mathsf{Vec}^{-1}(\mathbf{k}_j^*) \neq \mathsf{Vec}^{-1}(\mathbf{k}_k^*)$, but $\mathbf{t}_j^* = \mathbf{t}_k^* = \mathbf{t}^*$ and $\mathcal{B}$ could use $(\mathbf{k}_j^*, \mathbf{k}_k^*)$ with openings $(\mathbf{u}_j^*, \mathbf{u}_k^*)$ to break the binding of the commitment scheme. Furthermore, if $\alpha_j^* \neq \alpha_k^*$ (both from the ISIS_f challenger), then there exists some $j' \in [N+1] \setminus \{j,k\}$

such that $\alpha^*_{j'} = \alpha^*_j$ (or α^*_k) with $\mathsf{Vec}^{-1}(\mathbf{k}^*_{j'}) \neq \mathsf{Vec}^{-1}(\mathbf{k}^*_j)$ (or $\mathsf{Vec}^{-1}(\mathbf{k}^*_k)$) and the same argument as before applies.

On the other hand, if $\mathcal{A}$ *did not* receive at least one of α^*_j or α^*_k— without loss of generality, say α^*_j—from the $\mathcal{B}$, then $\mathbf{s}^\star := \left(\mathbf{s}^*_j - \mathbf{R}_0 \cdot \mathbf{k}^* - \mathbf{R}_1 \cdot \mathbf{u}^*\right)$ is a candidate ISIS_f solution. In fact, it is directly a solution if $\mathcal{B}$ never received α^*_j from the ISIS_f challenger. Moreover, in the case where $\mathcal{B}$ did receive α^*_j from the ISIS_f challenger (recall that this can happen if α^*_j was rejected during rejection sampling), $\mathcal{B}$ already has an $\mathbf{s}'$ satisfying $\mathbf{A} \cdot \mathbf{s}' = f(\alpha^*_j)$. As the adversary never directly saw $\mathbf{s}'$, and due to exponential entropy of preimage sampling (cf. §3.2), the probability that $\mathbf{s}^\star = \mathbf{s}'$ is negligible. Thus, in both cases, $\mathcal{B}$ outputs $\mathbf{s}^\star$.

Case 1.2 There exists some $j \in [NQ+1]$ such that $\mathcal{A}$ *did not* query O_{sk} for $(\mathbf{t}_j, \cdot, \cdot)$.

It follows by assumption that for all $k' \in [NQ+1]$,

$$\left(\begin{array}{c} \mathfrak{x}_{k'} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{k'}, \psi_{k'}), \\ \mathfrak{w}_{k'} := \left(\mathbf{k}_{k'}, \mathbf{u}_{k'}, \mathbf{s}_{k'}, \mathbf{g}^{-1}(\mathrm{i}_{k'}), \alpha_{k'}, \rho_{k'}\right) \end{array} \right) \in \mathscr{R}_2^{\mathsf{NIBBS},\mathsf{ISIS}_f} ,$$

so that, in particular, for some $j \in [NQ+1]$ with $(\mathbf{t}_j := \mathbf{C}_0 \cdot \mathbf{k}_j + \mathbf{C}_1 \cdot \mathbf{u}_j, \cdot, \cdot)$ not in the set of O_{sk} queries, we have $\mathbf{A}_j \cdot \mathbf{s}_j = \mathbf{C}_0 \cdot \mathbf{k}_j + \mathbf{C}_1 \cdot \mathbf{u}_j + f(\alpha_j)$. Now, suppose $\mathcal{A}$ received α_j from $\mathcal{B}$ (who in turn received it from the ISIS_f challenger) in the kth-query. For $\mathcal{B}$ to be successful, we then want that $\mathbf{s}^* = \mathbf{s}_j - \mathbf{R}_0 \cdot \mathbf{k}_j - \mathbf{R}_1 \cdot \mathbf{u}_j \neq \mathbf{s}^{(k)} = \mathbf{s}'^{(k)} - \mathbf{R}_0 \cdot \mathbf{k}^{(k)} - \mathbf{R}_1 \cdot \mathbf{u}^{(k)}$. Conversely, we want it to be hard for the adversary to come up with $\left(\mathbf{s}_j, \mathbf{k}_j, \mathbf{u}_j\right)$ such that

$$\left(\mathbf{s}_j - \mathbf{s}'^{(k)}\right) - \mathbf{R}_0 \cdot \left(\mathbf{k}_j - \mathbf{k}^{(k)}\right) - \mathbf{R}_1 \cdot \left(\mathbf{u}_j - \mathbf{u}^{(k)}\right) = \mathbf{0} \ . \tag{3}$$

Since $\mathbf{t}_j$ was never queried, by the condition of the current case, at least one of $\mathbf{k}_j \neq \mathbf{k}^{(k)}$ or $\mathbf{u}_j \neq \mathbf{u}^{(k)}$ must be true. We can now use the following lemma

Lemma B.5 ([LM18, Lemma 4.4])**.** *For n, m and q as defined above, for every $\mathbf{A} \in \mathbb{Z}^{n \times m}$, and $\mathbf{R} \leftarrow\!\!\$ \left\{\mathbf{U} \in \mathcal{R}^{m \times n' m_N} \ : \ \|\mathbf{U}\|_\infty \leqslant \widehat{\mathfrak{B}}\right\}$ where $\widehat{\mathfrak{B}} = \left\lceil \frac{q^{n/m} \cdot 2^{\lambda/m} - 1}{2} \right\rceil$, with probability at least $1 - t \cdot 2^{-\lambda}$, there exists an $\mathbf{Q} \in \mathcal{R}^{m \times n' m_N}$ from the same distribution as $\mathbf{R}$ such that $\mathbf{A} \cdot \mathbf{Q} = \mathbf{A} \cdot \mathbf{R}$ and, $\mathbf{Q}$ differs from $\mathbf{R}$ in t columns.*

to get that, if $\mathbf{k}_j \neq \mathbf{k}^{(k)}$, drawing $\mathbf{R}_0$ from the suitable distribution described above, with all but negligible probability, we are guaranteed a $\mathbf{Q}_0 \in \mathbb{Z}_2^{m \times n' m_N}$ from the same distribution that differs from $\mathbf{R}_0$ in a(ny) column where $\mathbf{k}_j$ and $\mathbf{k}^{(k)}$ differ. So, in particular, $\mathbf{R}_0 \cdot \left(\mathbf{x}_j - \mathbf{x}^{(k)}\right) \neq \mathbf{Q}_1 \cdot \left(\mathbf{k}_j - \mathbf{k}^{(k)}\right)$ but since $\mathbf{R}_0$ is hidden from $\mathcal{A}$, the probability of equation (3) being true is negligible. The same argument holds if $\mathbf{u}_j \neq \mathbf{u}^{(k)}$. Finally, if $\mathcal{A}$ never received α_j from $\mathcal{B}$ then by the same analysis as case (1.1), $\mathcal{B}$ has a solution for ISIS_f.

□

Proposition B.6. *An adversary has negligible advantage in case 2.*

Proof. Recall that the second case requires that the number of distinct α_j for $j \in [NQ+1]$ is exactly Q. Given, this we can further split our analysis into two distinct cases:

Case 2.1 There exists $j \in [Q]$ such that for the jth-query, with response $(\cdot, \alpha^{(j)})$, there are at least $N+1$ distinct corresponding messages $\{\mu_k\}_{k\in[N+1]}$ in $\mathcal{A}$'s final output.

It follows by assumption that for all $k' \in [NQ+1]$,

$$\left(\begin{array}{c} \mathfrak{x}_{k'} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{k'}, \psi_{k'}), \\ \mathfrak{w}_{k'} := \left(\mathbf{k}_{k'}, \mathbf{u}_{k'}, \mathbf{s}_{k'}, \mathbf{g}^{-1}(\mathrm{i}_{k'}), \alpha_{k'}, \rho_{k'}\right) \end{array}\right) \in \mathscr{R}_2^{\mathsf{NIBBS,ISIS}_f},$$

so that, in particular, it holds for all $k \in [N+1]$ that $\mu_k = \mathsf{F}_{\mathbf{K}^{(j)}}(\alpha^{(j)}, \mathrm{i}_k)$ and $\mathrm{i}_k \in [N]$. Furthermore, for the (one-more) forgery to be valid, all μ_k must be distinct and, therefore, all i_k must be distinct. Clearly this violates the soundness of NIZK_2, as we could then design a reduction $\mathcal{B}$ which specifically breaks the condition that $\mathrm{i}_k \in [N]$.

Case 2.2 For all $j \in [Q]$, there are exactly N distinct message-signature pairs corresponding to each query in $\mathcal{A}$'s final output.

In this case, the number of messages output by $\mathcal{A}$ must be NQ, which contradicts the assumption that $\mathcal{A}$ outputs more than NQ distinct messages.

□

It follows that an adversary of the second type also has negligible advantage.

□

Thus, the construction is one-more unforgeable. □

Lemma 6.4 (Inter-batch blindness). *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK_1 *and* NIZK_2 *are multi-theorem zero-knowledge proof systems and* PKE *is* IND-CPA *secure. Then, construction 6 is inter-batch blind.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBBS inter-batch blindness experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\$\ \mathbb{Z}_3^{\frac{m_N}{2}\times(m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\$\ \mathbb{Z}_3^{n\times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\$\ \mathbb{Z}_q^{n\times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$, $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$\ \mathsf{NIZK}_1.\mathsf{Setup}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\$\ \mathsf{NIZK}_2.\mathsf{Setup}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and sends it to the adversary.
2. For $b \in \{0,1\}$, the challenger samples the PRF key $\mathbf{K}_b \leftarrow\!\$\ \mathbb{Z}_2^{n'\times m_N}$, randomness $\mathbf{u}_b \leftarrow\!\$\ \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}_b$ to $\mathbf{k}_b := \mathsf{Vec}(\mathbf{K}_b) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t}_b := \mathbf{C}_0 \cdot \mathbf{k}_b + \mathbf{C}_1 \cdot \mathbf{u}_b$. Next, it generates a ciphertext, $\phi_b \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_b \| \mathbf{u}_b;\ \nu_b)$ from uniformly sampled randomness ν_b and the NIZK proof $\tau_b \leftarrow\!\$\ \mathsf{NIZK}_1.\mathsf{Prove}\big(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x}_b := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}_b, \phi_b), \mathfrak{w}_b := \big(\mathbf{k}_b, \mathbf{u}_b, \nu_b\big)$. It then sets $\mathsf{sk}_{R_b} := (\mathbf{K}_b, \mathbf{u}_b)$ and $\mathsf{pk}_{R_b} := (\mathbf{t}_b, \phi_b, \tau_b)$ and sends $\left(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}\right)$ to the adversary.

3. The adversary is allowed to make a series of $Q = \mathsf{poly}(\lambda)$ queries. In the jth query, it chooses $b^{(j)} \in \{0,1\}$, verification key $\mathsf{vk}^{(j)} := (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, pre-signature $\mathsf{psig}^{(j)} := \mathbf{s}^{(j)}$ and nonce $\mathsf{nonce}^{(j)} := \alpha^{(j)}$, and sends them as a signature obtain query to $O_{\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}}$. The challenger answers the jth query as follows:

   3.1 The challenger parses $\mathsf{sk}_{R_{b^{(j)}}}$ as $(\mathbf{K}_{b^{(j)}}, \mathbf{u}_{b^{(j)}})$ and then, for $\mathsf{vk} =: (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, it checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \alpha^{(j)} = \mathbf{C}_0 \cdot \mathbf{k}_{b^{(j)}} + \mathbf{C}_1 \cdot \mathbf{u}_{b^{(j)}}$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\alpha^{(j)} \in [2^m]$. If any check fails it aborts and outputs $\perp$.

   3.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{\mathsf{i}}^{(j)} := \mathbf{Y}_1 \cdot \left(\mathbf{K}_{b^{(j)}} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} f(\alpha^{(j)}) \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{\mathsf{i}}^{(j)} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_{b^{(j)}} \| \mathbf{u}_{b^{(j)}} \| \mathbf{s}^{(j)} \| \alpha^{(j)} \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{\mathsf{i}}^{(j)} \ \right)$$

   from uniformly sampled randomness $\rho_{\mathsf{i}}^{(j)}$. Then it generates the NIZK proof:

$$\pi_{\mathsf{i}}^{(j)} \leftarrow\!\!\$ \ \mathsf{NIZK}_2.\mathsf{Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_{\mathsf{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathsf{i}}^{(j)}, \psi_{\mathsf{i}}^{(j)} \right), \\ \mathfrak{w}_{\mathsf{i}}^{(j)} := \left(\mathbf{k}_{b^{(j)}}, \mathbf{u}_{b^{(j)}}, \mathbf{s}^{(j)}, \alpha^{(j)}, \mathbf{g}^{-1}(\mathsf{i}), \rho_{\mathsf{i}}^{(j)} \right) \end{array} \right),$$

   3.3 Finally, it outputs $\left[\!\left[\mu^{(j)}, \sigma^{(j)} \right]\!\right] := \{(\mu_1^{(j)}, \sigma_1^{(j)} := (\psi_1^{(j)}, \pi_1^{(j)})), \ldots, (\mu_N^{(j)}, \sigma_N^{(j)} := (\psi_N^{(j)}, \pi_N^{(j)}))\}$.

4. The adversary chooses and outputs $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and $(\mathsf{psig}_b := \mathbf{s}_b, \mathsf{nonce}_b := \alpha_b)$ for each $b \in \{0,1\}$. Then, for each $b \in \{0,1\}$, the challenger does the following:

   4.1 The challenger sets $\mathbf{s}_b =: \mathsf{psig}_b$ and $\alpha_b =: \mathsf{nonce}_b$ and then parses sk_{R_b} as $(\mathbf{K}_b, \mathbf{u}_b)$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, it checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \alpha_b = \mathbf{C}_0 \cdot \mathbf{k}_b + \mathbf{C}_1 \cdot \mathbf{u}_b$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\alpha_b \in [2^m]$. If any check fails it aborts and outputs $\perp$.

   4.2 Otherwise, for each $\mathsf{i} \in [N]$, the challenger sets

$$\mu_{b,\mathsf{i}} := \mathbf{Y}_1 \cdot \left(\mathbf{K}_b \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} f(\alpha_b) \\ \mathbf{g}^{-1}(\mathsf{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{b,\mathsf{i}} \leftarrow \mathsf{PKE.Enc}\left(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_b \| \mathbf{u}_b \| \mathbf{s}_b \| \alpha_b \| \mathbf{g}^{-1}(\mathsf{i}) \ ; \ \rho_{b,\mathsf{i}} \ \right)$$

   from uniformly sampled randomness $\rho_{b,\mathsf{i}}$. Then it generates the NIZK proof:

$$\pi_{b,\mathsf{i}} \leftarrow\!\!\$ \ \mathsf{NIZK}_2.\mathsf{Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_{b,\mathsf{i}} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathsf{i}}, \psi_{b,\mathsf{i}}), \\ \mathfrak{w}_{b,\mathsf{i}} := \left(\mathbf{k}_b, \mathbf{u}_b, \mathbf{s}_b, \alpha_b, \mathbf{g}^{-1}(\mathsf{i}), \rho_{b,\mathsf{i}} \right) \end{array} \right),$$

4.3 Finally, it outputs $\left(\llbracket \mu, \sigma \rrbracket_{\hat{b}}, \llbracket \mu, \sigma \rrbracket_{1-\hat{b}}\right)$ where $\llbracket \mu, \sigma \rrbracket_b := ((\mu_{b,1}, \sigma_{b,1} := (\psi_{b,1}, \pi_{b,1})), \ldots, (\mu_{b,N}, \sigma_{b,N} := (\psi_{b,N}, \pi_{b,N})))$ for each $b \in \{0,1\}$.

5. The (admissible) adversary outputs $b^* \in \{0,1\}$ and wins if $b^* = \hat{b}$.

Hybrid 1. This is the same as Hybrid 0 except that instead of honestly generating the NIZK proofs in steps 3 and 4, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using NIZK_2.Prove in answering the signing queries as well as computing the final signatures, it runs $\mathsf{NIZK}_2.\mathcal{S}$ without the witness.

Hybrid 0 and Hybrid 1 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.4.

Hybrid 2. This is the same as Hybrid 1 except that in step 4, instead of honestly computing the encryptions $\psi_{0,1}, \ldots, \psi_{0,N}$ on the secrets, it computes them as encryptions of $\mathbf{0}$.

Hybrid 1 and Hybrid 2 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.4.

Hybrid 3. This is the same as Hybrid 2 except that in step 4, instead of honestly computing the encryptions $\psi_{1,1}, \ldots, \psi_{1,N}$ on the secrets, it computes them as encryptions of $\mathbf{0}$.

The fact that Hybrids 2 and 3 are indistinguishable can be shown exactly as for Hybrids 1 and 2.

Hybrid 4. This is the same as Hybrid 3 except that instead of honestly generating the NIZK proof in step 2, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using NIZK_1.Prove to prove the commitments as part of the public keys, it runs $\mathsf{NIZK}_1.\mathcal{S}$ without the witness.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 3 and 4. We give a reduction algorithm $\mathcal{B}$ that breaks the multi-theorem zero-knowledge property of NIZK_1 with non-negligible advantage as follows:

- The NIZK challenger starts by sampling $b' \leftarrow\!\$ \{0,1\}$. If $b' = 0$, it samples $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$ \mathsf{NIZK}_1.\mathsf{Setup}(1^\lambda)$; otherwise, it samples $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$ \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$ and sends $\mathsf{crs}_{\mathsf{NIZK}_1}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$ \{0,1\}$. It then the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\$ \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\$ \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$ \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\$ \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 3), for each $b \in \{0,1\}$, the challenger samples the PRF key $\mathbf{K}_b \leftarrow\!\$ \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u}_b \leftarrow\!\$ \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}_b$ to $\mathbf{k}_b := \mathsf{Vec}(\mathbf{K}_b) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t}_b := \mathbf{C}_0 \cdot \mathbf{k}_b + \mathbf{C}_1 \cdot \mathbf{u}_b$. Next, it generates a ciphertext, $\phi_b \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_b \| \mathbf{u}_b;\ \nu_b)$ from uniformly sampled randomness ν_b and the NIZK proof $\tau_b \leftarrow\!\$ \mathsf{NIZK}_1.\mathsf{Prove}\big(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x}_b := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}_b, \phi_b), \mathfrak{w}_b := \big(\mathbf{k}_b, \mathbf{u}_b, \nu_b\big)$. It then sets $\mathsf{sk}_{R_b} := (\mathbf{K}_b, \mathbf{u}_b)$ and $\mathsf{pk}_{R_b} := (\mathbf{t}_b, \phi_b, \tau_b)$ and sends $\left(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}\right)$ to the adversary.
- The rest of the reduction proceeds according to Hybrid 3.

- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received an honestly generated proof from NIZK challenger; otherwise, it outputs 1.

If challenger uses a NIZK simulator, $\mathcal{B}$ perfectly simulates Hybrid 4 to $\mathcal{A}$; otherwise, it simulates Hybrid 3. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the NIZK zero-knowledge challenger.

Hybrid 5. This is the same as Hybrid 4 except that in steps 2, instead of honestly computing the encryption ϕ_0, it computes it as as encryptions of $\mathbf{0}$.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 4 and 5. We give a reduction algorithm $\mathcal{B}$ that breaks the IND-CPA security of PKE with non-negligible advantage as follows:

- The IND-CPA challenger chooses a bit $b' \leftarrow\!\!\$\ \{0,1\}$ and samples $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\ \mathsf{PKE.Setup}(1^\lambda)$ and sends $\mathsf{pk}_{\mathsf{PKE}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\!\$\ \{0,1\}$. It then the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\!\$\ \mathbb{Z}_3^{\frac{m_N}{2} \times (m+m_N-m')}$, $\mathbf{Y}_1 \leftarrow\!\!\$\ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times n' m_N}$ along with $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\!\$\ \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\!\$\ \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and sends it to the adversary.
- For $b \in \{0,1\}$, the challenger samples the PRF key $\mathbf{K}_b \leftarrow\!\!\$\ \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u}_b \leftarrow\!\!\$\ \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}_b$ to $\mathbf{k}_b := \mathsf{Vec}(\mathbf{K}_b) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t}_b := \mathbf{C}_0 \cdot \mathbf{k}_b + \mathbf{C}_1 \cdot \mathbf{u}_b$. Next, it generates a ciphertext, $\phi_1 \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}_1 \| \mathbf{u}_1;\ \nu_1)$ from uniformly sampled randomness ν_1 but instead of computing ϕ_0 the same way, it sets $\mathbf{m}_0 := \mathbf{k}_0 \| \mathbf{u}_0$ and $\mathbf{m}_1 := \mathbf{0}$ and sends $(\mathbf{m}_0, \mathbf{m}_1)$ to the challenger who responds with a ciphertext ϕ_0 (to $\mathbf{m}_{b'}$). It then simulates the NIZK proofs $\tau_b \leftarrow\!\!\$\ \mathsf{NIZK}_1.\mathcal{S}(\mathfrak{x}_b := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}_b, \phi_b))$. It sets $\mathsf{sk}_{R_b} := (\mathbf{K}_b, \mathbf{u}_b)$ and $\mathsf{pk}_{R_b} := (\mathbf{t}_b, \phi_b, \tau_b)$ and sends $\left(\mathsf{pk}_{R_0}, \mathsf{pk}_{R_1}\right)$ to the adversary.
- The rest of the protocol proceeds exactly as Hybrid 4 so that $\mathcal{B}$ sends the resulting $\left([\![\mu, \sigma]\!]_{\hat{b}}, [\![\mu, \sigma]\!]_{1-\hat{b}}\right)$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received a ciphertext of R_0's secrets from the IND-CPA challenger; otherwise, it outputs 1.

If challenger encrypts $\mathbf{m}_0$, $\mathcal{B}$ perfectly simulates Hybrid 4 to $\mathcal{A}$; otherwise, it simulates Hybrid 5. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the IND-CPA challenger.

Hybrid 6. This is the same as Hybrid 5 except that in steps 2, instead of honestly computing the encryption ϕ_1, it computes it as as encryptions of $\mathbf{0}$.

The fact that Hybrids 5 and 6 are indistinguishable can be shown exactly as for Hybrids 4 and 5.

Hybrid 7. This is the same as Hybrid 6 except that in step 2, instead computing a commitment $\mathbf{t}_0 := \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_0) + \mathsf{H}(\mathbf{u}_0)$ to $\mathbf{K}_0$, the challenger samples $\mathbf{t}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^n$.

Hybrid 6 and Hybrid 7 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.4.

Hybrid 8. This is the same as Hybrid 7 except that in step 2, instead computing a commitment $\mathbf{t}_1 := \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}_1) + \mathsf{H}(\mathbf{u}_1)$ to $\mathbf{K}_1$, the challenger samples $\mathbf{t}_1 \leftarrow\!\!\$\ \mathbb{Z}_q^n$.

The fact that Hybrids 7 and 8 are statistically indistinguishable can be shown exactly as it was for Hybrids 6 and 7.

Hybrid 9. This is the same as Hybrid 8 except that instead of computing the messages for receiver R_0 as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

Hybrid 8 and Hybrid 9 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.4.

Hybrid 10. This is the same as Hybrid 9 except that instead of computing the messages for receiver R_1 as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

The fact that Hybrids 9 and 10 are indistinguishable can be shown exactly as for Hybrids 8 and 9.

Note that any adversary has 0 advantage in Hybrid 10. Thus, the construction is inter-batch blind. □

Lemma 6.5 (Intra-batch blindness). *Suppose the* $\mathsf{F}_{\mathbf{K}}$ *is a pseudorandom function,* NIZK_1 *and* NIZK_2 *are multi-theorem zero-knowledge proof systems and* PKE *is* IND-CPA *secure. Then, construction 6 is intra-batch blind.*

Proof. The proof proceeds through a sequence of hybrids.

Hybrid 0. This is the original NIBS intra-batch blindness experiment.

1. Given input $(n, m, n', m', \mathfrak{s})$ the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\!\$\ \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\!\$\ \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and sends it to the adversary.

2. The challenger first chooses $\hat{b} \leftarrow\!\!\$\ \{0,1\}$. It then samples the PRF key $\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u} \leftarrow\!\!\$\ \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}$ to $\mathbf{k} := \mathsf{Vec}(\mathbf{K}) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t} := \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$. Next, it generates a ciphertext, $\phi \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u};\ \nu)$ from uniformly sampled randomness ν and the NIZK proof $\tau \leftarrow\!\!\$\ \mathsf{NIZK}_1.\mathsf{Prove}\big(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x} := (\mathbf{C}_0, \mathbf{R}_0, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}, \phi), \mathfrak{w} := \big(\mathbf{k}, \mathbf{u}, \nu\big)$. It then sets $\mathsf{sk}_R := (\mathbf{K}, \mathbf{u})$ and $\mathsf{pk}_R := (\mathbf{t}, \phi, \tau)$ and sends pk_R to the adversary.

3. The adversary is allowed to make a series of $Q = \mathsf{poly}(\lambda)$ queries. In the jth query, it chooses verification key $\mathsf{vk}^{(j)} := (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, pre-signature $\mathsf{psig}^{(j)} := \mathbf{s}^{(j)}$ and nonce $\mathsf{nonce}^{(j)} := \alpha^{(j)}$, and sends them as a signature obtain query to O_{pk_R}. The challenger answers the jth query as follows:

   3.1 The challenger parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk}^{(j)} =: (\mathbf{A}^{(j)}, \mathbf{B}^{(j)})$, it checks if $\mathbf{A}^{(j)} \cdot \mathbf{s}^{(j)} + \mathbf{B}^{(j)} \cdot \alpha^{(j)} = \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$, $\left\|\mathbf{s}^{(j)}\right\| \leqslant \mathfrak{B}$ and $\alpha^{(j)} \in [2^m]$. If any check fails it aborts and outputs $\perp$.

   3.2 Otherwise, for each $\mathfrak{i} \in [N]$, the challenger sets

$$\mu_{\mathfrak{i}}^{(j)} := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} f(\alpha^{(j)}) \\ \mathbf{g}^{-1}(\mathfrak{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \,.$$

With this, it now generates the ciphertext over its secrets as

$$\psi_{\mathrm{i}}^{(j)} \leftarrow \mathsf{PKE.Enc}\Big(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}\|\mathbf{u}\|\mathbf{s}^{(j)}\|\alpha^{(j)}\|\mathbf{g}^{-1}(\mathrm{i}) \ ; \ \rho_{\mathrm{i}}^{(j)} \ \Big)$$

from uniformly sampled randomness $\rho_{\mathrm{i}}^{(j)}$. Then it generates the NIZK proof:

$$\pi_{\mathrm{i}}^{(j)} \leftarrow\!\$ \ \mathsf{NIZK}_2.\mathsf{Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_{\mathrm{i}}^{(j)} := \left(\mathbf{A}^{(j)}, \mathbf{B}^{(j)}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{\mathrm{i}}^{(j)}, \psi_{\mathrm{i}}^{(j)}\right), \\ \mathfrak{w}_{\mathrm{i}}^{(j)} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}^{(j)}, \alpha^{(j)}, \mathbf{g}^{-1}(\mathrm{i}), \rho_{\mathrm{i}}^{(j)}\right) \end{array} \right),$$

3.3 Finally, it outputs $\left[\!\left[\mu^{(j)}, \sigma^{(j)}\right]\!\right] := ((\mu_1^{(j)}, \sigma_1^{(j)} := (\psi_1^{(j)}, \pi_1^{(j)})), \ldots, (\mu_N^{(j)}, \sigma_N^{(j)} := (\psi_N^{(j)}, \pi_N^{(j)})))$.

4. The adversary chooses and outputs $\mathsf{vk} := (\mathbf{A}, \mathbf{B})$ and $(\mathsf{psig}_b := \mathbf{s}_b, \mathsf{nonce}_b := \alpha_b)$ for each $b \in \{0,1\}$, as well as a permutation function $\varphi : [N] \times \{0,1\} \to [N] \times \{0,1\}$ for each $b \in \{0,1\}$. Then, for each $b \in \{0,1\}$, the challenger does the following:

   4.1 The challenger sets $\mathbf{s}_b =: \mathsf{psig}_b$ and $\alpha_b =: \mathsf{nonce}_b$ and then parses sk_R as $(\mathbf{K}, \mathbf{u})$ and then, for $\mathsf{vk} =: (\mathbf{A}, \mathbf{B})$, it checks if $\mathbf{A} \cdot \mathbf{s}_b + \mathbf{B} \cdot \alpha_b = \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$, $\|\mathbf{s}_b\| \leqslant \mathfrak{B}$ and $\alpha_b \in [2^m]$. If any check fails it aborts and outputs $\perp$.

   4.2 Otherwise, for each $\mathrm{i} \in [N]$, the challenger sets

$$\mu_{b,\mathrm{i}} := \mathbf{Y}_1 \cdot \left(\mathbf{K} \cdot \mathbf{G}^{-1} \left(\mathbf{Y}_0 \cdot \begin{bmatrix} \alpha \\ \mathbf{g}^{-1}(\mathrm{i}) \end{bmatrix} \bmod 3 \right) \bmod 2 \right) \bmod 3 \, .$$

   With this, it now generates the ciphertext over its secrets as

$$\psi_{b,\mathrm{i}} \leftarrow \mathsf{PKE.Enc}\Big(\ \mathsf{pk}_{\mathsf{PKE}}, \mathbf{k}\|\mathbf{u}\|\mathbf{s}_b\|\alpha_b\|\mathbf{g}^{-1}(\mathrm{i}) \ ; \ \rho_{b,\mathrm{i}} \ \Big)$$

   from uniformly sampled randomness $\rho_{b,\mathrm{i}}$. Then it generates the NIZK proof:

$$\pi_{b,\mathrm{i}} \leftarrow\!\$ \ \mathsf{NIZK}_2.\mathsf{Prove}\left(\begin{array}{c} \mathsf{crs}_{\mathsf{NIZK}_2}, \mathfrak{x}_{b,\mathrm{i}} := (\mathbf{A}, \mathbf{B}, \mathbf{C}_0, \mathbf{C}_1, \mathbf{Y}_0, \mathbf{Y}_1, f, \mathsf{pk}_{\mathsf{PKE}}, \mu_{b,\mathrm{i}}, \psi_{b,\mathrm{i}}), \\ \mathfrak{w}_{b,\mathrm{i}} := \left(\mathbf{k}, \mathbf{u}, \mathbf{s}_b, \alpha_b, \mathbf{g}^{-1}(\mathrm{i}), \rho_{b,\mathrm{i}}\right) \end{array} \right),$$

   4.3 Finally, if $\hat{b} = 0$, it outputs the sequence $\{(\mu_{\mathrm{i},b}, \sigma_{\mathrm{i},b})\}_{\substack{\mathrm{i}\in[N] \\ b\in\{0,1\}}}$. Otherwise, it outputs $\left\{\left(\mu_{\varphi(\mathrm{i},b)}, \sigma_{\varphi(\mathrm{i},b)}\right)\right\}_{\substack{\mathrm{i}\in[N] \\ b\in\{0,1\}}}$.

5. The (admissible) adversary outputs $b^* \in \{0,1\}$ and wins if $b^* = \hat{b}$.

Hybrid 1. This is the same as Hybrid 0 except that instead of honestly generating the NIZK proofs in steps 3 and 4, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using $\mathsf{NIZK}_2.\mathsf{Prove}$ in answering the signing queries as well as computing the final signatures, it runs $\mathsf{NIZK}_2.\mathcal{S}$ without the witness.

Hybrid 0 and Hybrid 1 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.5.

Hybrid 2. This is the same as Hybrid 1 except that in step 4, instead of honestly computing the encryptions $\psi_{0,1}, \ldots, \psi_{0,N}$ on the secrets, it computes them as encryptions of $\mathbf{0}$.

Hybrid 1 and Hybrid 2 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.5.

Hybrid 3. This is the same as Hybrid 2 except that in step 4, instead of honestly computing the encryptions $\psi_{1,1}, \ldots, \psi_{1,N}$ on the secrets, it computes them as encryptions of $\mathbf{0}$.

The fact that Hybrids 2 and 3 are indistinguishable can be shown exactly as for Hybrids 1 and 2.

Hybrid 4. This is the same as Hybrid 3 except that instead of honestly generating the NIZK proof in step 2, it runs the NIZK simulator and outputs a simulated proof. More precisely, it runs $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$ and instead of computing the proofs using $\mathsf{NIZK}_1.\mathsf{Prove}$ to prove the commitments as part of the public keys, it runs $\mathsf{NIZK}_1.\mathcal{S}$ without the witness.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 3 and 4. We give a reduction algorithm $\mathcal{B}$ that breaks the multi-theorem zero-knowledge property of NIZK_1 with non-negligible advantage as follows:

- The NIZK challenger starts by sampling $b' \leftarrow\!\$\, \{0,1\}$. If $b' = 0$, it samples $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$\, \mathsf{NIZK}_1.\mathsf{Setup}(1^\lambda)$; otherwise, it samples $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\$\, \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$ and sends $\mathsf{crs}_{\mathsf{NIZK}_1}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\$\, \{0,1\}$. It then the challenger samples matrices $\mathbf{Y}_0 \leftarrow\!\$\, \mathbb{Z}_3^{\frac{m_N}{2} \times (m + m_N - m')}$, $\mathbf{Y}_1 \leftarrow\!\$\, \mathbb{Z}_3^{n \times n'}$ and $\mathbf{C}_0, \mathbf{C}_1 \leftarrow\!\$\, \mathbb{Z}_q^{n \times n' m_N}$ along with $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\, \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\$\, \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$. It sets $\mathsf{pp} := (1^\lambda, n, m, n', m_N, \mathfrak{s}, q, N, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2}, \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{C}_0, \mathbf{C}_1)$ and sends it to the adversary.
- Similarly to step 2 (Hybrid 3), the challenger samples the PRF key $\mathbf{K} \leftarrow\!\$\, \mathbb{Z}_2^{n' \times m_N}$, randomness $\mathbf{u} \leftarrow\!\$\, \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}$ to $\mathbf{k} := \mathsf{Vec}(\mathbf{K}) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t} := \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$. Next, it generates a ciphertext, $\phi \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{k} \| \mathbf{u}; \nu)$ from uniformly sampled randomness ν and the NIZK proof $\tau \leftarrow\!\$\, \mathsf{NIZK}_1.\mathsf{Prove}\big(\mathsf{crs}_{\mathsf{NIZK}_1}, \mathfrak{x} := (\mathbf{C}_0, \mathbf{R}_0, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}, \phi), \mathfrak{w} := \big(\mathbf{k}, \mathbf{u}, \nu\big)$. It then sets $\mathsf{sk}_R := (\mathbf{K}, \mathbf{u})$ and $\mathsf{pk}_R := (\mathbf{t}, \phi, \tau)$ and sends pk_R to the adversary.
- The rest of the reduction proceeds according to Hybrid 3.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received an honestly generated proof from NIZK challenger; otherwise, it outputs 1.

If challenger uses a NIZK simulator, $\mathcal{B}$ perfectly simulates Hybrid 4 to $\mathcal{A}$; otherwise, it simulates Hybrid 3. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the NIZK zero-knowledge challenger.

Hybrid 5. This is the same as Hybrid 4 except that in steps 2, instead of honestly computing the encryption ϕ, it computes it as as encryptions of $\mathbf{0}$.

Suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between Hybrids 4 and 5. We give a reduction algorithm $\mathcal{B}$ that breaks the IND-CPA security of PKE with non-negligible advantage as follows:

- The IND-CPA challenger chooses a bit $b' \leftarrow\!\!\$\ \{0,1\}$ and samples $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\ \mathsf{PKE.Setup}(1^\lambda)$ and sends $\mathsf{pk}_{\mathsf{PKE}}$ to $\mathcal{B}$.
- $\mathcal{B}$ first chooses $\hat{b} \leftarrow\!\!\$\ \{0,1\}$ and then samples $v_j \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R},\mathfrak{s}_v}$ for every $j \in [N]$ and $z_j \leftarrow\!\!\$\ \mathcal{D}_{\mathcal{R},\mathfrak{s}_v}$ for every $j \in [B]$, $\mathbf{R}_1 \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m'}$, $\mathbf{R}_2 \leftarrow\!\!\$\ \mathcal{R}_q^{n\times \ell}$, $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\!\$\ \mathsf{PKE.KeyGen}(1^\lambda)$, $\mathsf{crs}_{\mathsf{NIZK}_1} \leftarrow\!\!\$\ \mathsf{NIZK}_1.\mathcal{S}(1^\lambda)$, $\mathsf{crs}_{\mathsf{NIZK}_2} \leftarrow\!\!\$\ \mathsf{NIZK}_2.\mathcal{S}(1^\lambda)$, sets $\mathsf{pp} := (B, m, m', n, N, p, q, d, f, \kappa, \mathfrak{s}_v, \mathfrak{s}_{\mathbf{s}}, \mathfrak{s}_{\mathbf{x}}, \mathbf{R}_1, \mathbf{R}_2, \mathsf{pk}_{\mathsf{PKE}}, \{v_j\}, \{z_j\}, \mathsf{crs}_{\mathsf{NIZK}_1}, \mathsf{crs}_{\mathsf{NIZK}_2})$ and sends it to the adversary.
- The challenger samples the PRF key $\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_2^{n'\times m_N}$, randomness $\mathbf{u} \leftarrow\!\!\$\ \mathbb{Z}_2^{n' m_N}$, and then computes a commitment $\mathbf{t}$ to $\mathbf{k} := \mathsf{Vec}(\mathbf{K}) \in \mathbb{Z}_2^{n' m_N}$ as $\mathbf{t} := \mathbf{C}_0 \cdot \mathbf{k} + \mathbf{C}_1 \cdot \mathbf{u}$. Next, instead of computing ϕ the same way, it sets $\mathbf{m}_0 := \mathbf{k} \| \mathbf{u}$ and $\mathbf{m}_1 := \mathbf{0}$ and sends $(\mathbf{m}_0, \mathbf{m}_1)$ to the challenger who responds with a ciphertext ϕ (to $\mathbf{m}_{b'}$). It then simulates the NIZK proofs $\tau \leftarrow\!\!\$\ \mathsf{NIZK}_1.\mathcal{S}(\mathfrak{x} := (\mathbf{C}_0, \mathbf{C}_1, \mathsf{pk}_{\mathsf{PKE}}, \mathbf{t}, \phi))$. It sets $\mathsf{sk}_R := (\mathbf{K}, \mathbf{u})$ and $\mathsf{pk}_R := (\mathbf{t}, \phi, \tau)$ and sends pk_R to the adversary.
- The rest of the protocol proceeds exactly as Hybrid 4 so that if $\hat{b} = 0$, $\mathcal{B}$ send the sequence $\{(\mu_{\mathrm{i},b}, \sigma_{\mathrm{i},b})\}_{\substack{\mathrm{i}\in[B] \\ b\in\{0,1\}}}$; and otherwise sends $\left\{\left(\mu_{\varphi(\mathrm{i},b)}, \sigma_{\varphi(\mathrm{i},b)}\right)\right\}_{\substack{\mathrm{i}\in[B] \\ b\in\{0,1\}}}$ to $\mathcal{A}$.
- Finally, when $\mathcal{A}$ outputs b^*, $\mathcal{B}$ outputs 0 if $b^* = \hat{b}$ indicating its guess that it received a ciphertext of R's secrets from the IND-CPA challenger; otherwise, it outputs 1.

If challenger encrypts $\mathbf{m}_0$, $\mathcal{B}$ perfectly simulates Hybrid 4 to $\mathcal{A}$; otherwise, it simulates Hybrid 5. Thus, if $\mathcal{A}$ can distinguish the two hybrids with non-negligible advantage, then $\mathcal{B}$ has non-negligible advantage against the IND-CPA challenger.

Hybrid 6. This is the same as Hybrid 5 except that in step 2, instead computing a commitment $\mathbf{t} := \mathbf{C} \cdot \mathsf{Vec}(\mathbf{K}) + \mathsf{H}(\mathbf{u})$ to $\mathbf{K}$, the challenger samples $\mathbf{t}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^n$.

Hybrid 5 and Hybrid 6 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.5.

Hybrid 7. This is the same as Hybrid 6 except that instead of computing the messages for receiver R as a PRF evaluation in steps 3 and 4, it samples it uniformly from $\mathbb{Z}_3^n$.

Hybrid 6 and Hybrid 7 are indistinguishable by the same reasoning as shown in the proof of Lemma 5.5.

Note that any adversary has 0 advantage in Hybrid 7. Thus, the construction is intra-batch blind. □

This completes the proof of the theorem.