Improved Lattice Blind Signatures from Recycled Entropy

Corentin Jeudy[1] and Olivier Sanders[1]

corentin.jeudy@orange.com, olivier.sanders@orange.com

Orange Labs, Applied Crypto Group, Cesson-Sévigné, France

Abstract. Blind signatures represent a class of cryptographic primitives enabling privacy-preserving authentication with several applications such as e-cash or e-voting. It is still a very active area of research, in particular in the post-quantum setting where the history of blind signatures has been hectic. Although it started to shift very recently with the introduction of a few lattice-based constructions, all of the latter give up an important characteristic of blind signatures (size, efficiency, or security under well-known assumptions) to achieve the others. In this paper, we propose another design which revisits the link between the two main procedures of blind signatures, namely issuance and showing, demonstrating that we can significantly alleviate the second one by adapting the former. Concretely, we show that we can harmlessly inject excess randomness in the issuance phase, and then recycle the entropy surplus during showing to decrease the complexity of the zero-knowledge proof which constitutes the main component of the signature. This leads to a blind signature scheme with small sizes, low complexity, and that still relies on well-known lattice assumptions.

Keywords: Lattice-Based Cryptography · Blind Signature · Privacy

1 Introduction

Introduced by Chaum in 1982, blind signatures [Cha82] are one of the fundamental tools of cryptography for privacy. Contrarily to standard signatures where the signer knows the message m they sign and can trace the signature s they generate in the process, blind signatures aim at hiding both m and s to the signer so as to limit traceability. More precisely, a blind signature comes with an interactive signing protocol where one can get a signature s on a message m such that (1) validity of s can be publicly verified (as is the case with standard signatures) and (2) the signer cannot link (s, m) to a specific issuance. The latter property necessarily implies that m is hidden to the signer at the issuance time, hence the term "*blind signature*".

This feature might seem unwise for those who have classical signature use-cases in mind (e.g., signing a contract) but there are some other applications where blind signatures significantly improve privacy without jeopardizing security. This is typically the case of electronic cash that was introduced by Chaum in his seminal work mentioned above. In this context, the signed "message" is essentially a serial number of a coin whose only purpose is to facilitate double-spending detection. For the signer (the "bank" in the e-cash terminology), the only value that matters is the number of signatures it issues as it corresponds to the number of minted coins, not their actual serial numbers. Blinding the latter is then harmless to security while being essential for privacy. Similar situations occur in other use-cases such as electronic voting [BW16], which led ISO/IEC to standardize blind signature [ISO16].

In [Cha83], Chaum introduced a first construction which astutely leverages the RSA signing function. It defines a 2-round protocol where the user first transmits a blinded message $c = r^e \cdot H(m) \bmod N$ where N is the RSA modulus, e is the signer's public key and r is a random element of $\mathbb{Z}_N$. The signer then generates an element s' such that $(s')^e = c \bmod N$ which can be unblinded by computing $s = s'/r$. Indeed, $s^e = H(m) \bmod N$, meaning that s is a valid RSA signature on m. This elegant design has however a downside as no security reduction to a standard assumption is known as of today. This led Bellare et al. [BNPS03] to introduce a tailored interactive assumption, one-more-RSA, where the adversary has access to an RSA inverse oracle. Thanks to this oracle, the reduction can easily handle all the signing queries of the EUF-CMA security experiment and then extract the "one-more" RSA inverse from the forgery. It thus allows the reduction to succeed but the security assurances such an assumption provides are arguably lower than those provided by a standard computational assumption.

Another approach, based on the celebrated Schnorr identification protocol [Sch89], was later proposed, with several variants, some of which proved under the discrete logarithm assumption (see [PS00] and references therein). All these schemes inherit the 3-round approach of Schnorr's sigma protocol and thus require an additional interaction compared to [Cha83]. While the difference might seem insignificant at a time of widespread fast communications, the sigma protocol approach introduces a very subtle issue when several signatures are generated in parallel. This problem was already detected by Pointcheval and Stern [PS00] who noticed that security can only be proven when a very low limit is enforced on the number of concurrent executions of the protocols. Far from being an artefact of the proof, this problem stems from a very concrete vulnerability studied by Wagner [Wag02]. The resulting ROS attack, later improved in [BLL$^+$22] demonstrates the limits of Schnorr's blind signature and its variants. This has led cryptographers to favour two-round designs although we stress that this problem is not inherent to 3-round blind signatures, only to those following Schnorr's blueprint.

The development of efficient zero-knowledge proof systems has led to an alternative approach in the design of blind signatures which was formalized by Fischlin [Fis06]. It essentially consists in (1) issuing a standard signature s on a hiding commitment c to the message m and (2) producing a zero-knowledge proof that s is valid for the committed m. This framework actually shares many commonalities with the ones of other privacy-preserving primitives such as group signature [CvH91,BMW03], direct anonymous attestation [BCC04] or anonymous credentials [CL01] and it is thus natural that all these mechanisms can be implemented with the same building blocks, namely zero-knowledge proofs and so-called signatures with efficient protocols (SEP) [CL02,CL04,BB08,PS16].

Unsurprisingly, the development of post-quantum blind signatures has proved harder. To our knowledge, the first construction was proposed by Ruckert [Rüc10], and was later improved in [ABB20,BCE+20]. These schemes were based on Lyubashevsky's identification protocol [Lyu08] which can somehow be seen as the lattice counterpart of Schnorr's protocol. As a consequence, they inherit the limited number of allowed parallel issuances. Worse, these constructions were proven unsafe by Hauck et al. [HKLN20] who pointed out errors in their security proofs and even attacks for some of them. In the same paper, the authors proposed a sound construction but with poor performance as blind signatures are 7.73 MB long even when the total number of signatures is limited to 7.

In [AKSY22], the authors propose an interesting mix between the original construction of Chaum and the framework of Fischlin. Concretely, the user first generates a commitment $\mathbf{c} = \mathbf{A}\mathbf{r} + H(m)$ which is inverted by the signer who generates a short $\mathbf{s}$ such that $\mathbf{C}\mathbf{s} = \mathbf{c}$ ($\mathbf{A}$ and $\mathbf{C}$ are public elements and $\mathbf{r}$ is a short random vector). A proof of knowledge of $\mathbf{s}$ is then produced, as in [Fis06]. The first step of the protocol can thus be seen as a transposition of Chaum's approach in the lattice setting, where $H(m)$ is blinded using $\mathbf{A}\mathbf{r}$ (instead of r^e) and then inverted. Unfortunately, it leads to the same problem as in [Cha83]: the reduction is unable to answer signing queries as it does not know any trapdoor for $\mathbf{C}$ in the security game. This led [AKSY22] to resort to the same trick as in [BNPS03], namely introduce a tailored interactive assumption called one-more-ISIS. The nice performance of [AKSY22] comes thus at the cost of a heuristic security.

In [dPK22], del Pino and Katsumata circumvent this problem by using the commitment to the message as a tag in a tag-based trapdoor system akin to [MP12]. Thanks to this solution, they were able to prove security of their construction under standard assumptions but at the cost of larger signatures (about 100 KB per blind signature).

Very recently, Beullens et al. [BLNS23a] revisited the first part of the construction in [AKSY22] by generating the commitment $\mathbf{c}$ as $\mathbf{A}\mathbf{r} + H(H(\mathbf{r}), m)$. This apparently simple modification allows to prove security of the resulting construction under standard assumptions while offering a very attractive signature size (22 KB). Unfortunately, the issuance process requires a proof that $\mathbf{c}$ is well-formed, which is not easy to produce given its reliance on hash functions. The authors suggest to use general purpose NIZKs but the latter are very complex to implement and are very long to generate (about 20 seconds per proof according to the authors' estimation), which is likely to affect users' experience.

Overall, when looking at the most advanced post-quantum blind signature schemes from the state-of-the-art, one must then choose between an efficient construction with heuristic security [AKSY22], a short and secure blind signature but with an impractical issuance process [BLNS23a] and a secure construction with larger signatures [dPK22]. As also highlighted in the context of e-cash in a project by the Swiss national bank [HB23], there is a need for future research and development on post-quantum blind signatures.

1.1 Our Contributions

In this paper, we introduce a new lattice-based blind signature construction which is both efficient and proven secure under standard assumptions. In particular, it comes with practical issuance and verification protocols, as demonstrated by our implementation, and our signatures sizes are slightly smaller than those in [AKSY22], namely 40 KB.

Our starting point is the first step of 2-round blind signature issuance protocols. We note that, for all of them, it consists in generating some commitment $\mathbf{c}$ to the (hashed) message $\mathcal{H}(m)$ blinded with some mask depending on a random element $\mathbf{r}$. In the context of cyclic or RSA groups, this mask can usually be removed at the end of the process as illustrated by Chaum's protocol [Cha83] or the ones we could build using some SEP schemes (e.g., [PS16]). In such a case, $\mathbf{r}$ does not affect the size of the blind signature and this approach therefore seems almost optimal. This is no longer true for lattice constructions [AKSY22,dPK22,BLNS23a] which must carry this randomness into the signature where it is included in the witnesses of the NIZK proof. Although the performance of lattice zero-knowledge proofs have significantly improved over the past few years, as illustrated by the powerful framework of [LNP22], the complexity of the latter is still impacted by (1) the dimension of the witnesses vector and (2) the size of the witnesses as larger witnesses requires larger masks and in turn a larger modulus. As a consequence, the papers above naturally choose the minimal parameters for $\mathbf{r}$ so as to ensure concealment of $\mathcal{H}(m)$ (through a mask $\mathbf{Ar}$) while minimizing the impact on the proof size.

In this paper, we take the opposite view. We choose a dimension and parameters for $\mathbf{r}$ which are much larger than what is necessary to hide $\mathcal{H}(m)$ in the issuance process, but then recycle the remaining entropy to directly mask large parts of the preimage $\mathbf{s}$ so as to reduce the whole witness size in the NIZK proof. As we will explain, we can design our protocol in such a way that the negative impact of larger $\mathbf{r}$ is largely offset by the positive impact of smaller NIZK proofs.

Let us consider the following basic blind signature scheme where the user first blinds the message $\mathcal{H}(m)$ by computing a commitment $\mathbf{c} = \mathbf{Ar} + \mathbf{d}\mathcal{H}(m)$ (for some public $\mathbf{d}$) and then obtain a short $\mathbf{s}'$ from the signer such that $\mathbf{Cs}' = \mathbf{c}$ for a public matrix $\mathbf{C}$. Our first remark is that, by enforcing mild requirements on $\mathbf{C}$, one can set $\mathbf{A} = \mathbf{C}$ without jeopardizing blindness. In such a case, we usually end up with a vector $\mathbf{r}$ of higher dimension (because $\mathbf{A}$ has generally less columns than $\mathbf{C}$) but we can then merge $\mathbf{s}'$ and $\mathbf{r}$ since we now have $\mathbf{A}(\mathbf{s}' - \mathbf{r}) = \mathbf{d}\mathcal{H}(m)$. At this stage, compared to the classical approach, we thus only need a proof of knowledge of a unique vector $(\mathbf{s}' - \mathbf{r})$ instead of two vectors $\mathbf{s}'$ and $\mathbf{r}$, which already reduces the proof size.

However, when we look at the new relation $\mathbf{A}(\mathbf{s}' - \mathbf{r}) = \mathbf{d}\mathcal{H}(m)$, we can make the following comments. The first one is that the size of $\mathbf{r}$ moderately impacts the one of $\mathbf{s} = (\mathbf{s}' - \mathbf{r})$ as long as the coefficients of $\mathbf{r}$ remain smaller than those of $\mathbf{s}'$. In other words, trying to make $\mathbf{r}$ as small as possible has only very limited impacts on the resulting vector $\mathbf{s}$. The second one is that our approach leads $\mathbf{r}$ to play two roles. As in previous post-quantum blind signatures schemes, it is still the key to blind m. But now, it also acts as some perturbation of the preimage $\mathbf{s}'$ which is very much akin to the one used in the preimage sampling process of [MP12]. While we cannot use the convolution results of [Pei10] to conclude in our case, we can still recycle a large part of the entropy of $\mathbf{r}$ to blind parts of $\mathbf{s}'$. More precisely, we show in our paper that we can fully disclose the lower part (i.e., the lower bits) of each element in $\mathbf{s}$ as they are perfectly blinded by the remaining entropy of $\mathbf{r}$. It thus only remains to hide the higher part of $\mathbf{s}$ in a zero-knowledge proof whose size is significantly smaller than the one of the classical approach where the whole vector $\mathbf{s}$ would have been hidden. Obviously the amount of bits of $\mathbf{s}$ we can safely leak directly depends on the size of $\mathbf{r}$ but, as mentioned in our first remark, we can harmlessly increase the latter as long as we remain under a reasonable threshold. In other words, we rely on the somewhat counter-intuitive reasoning : the more we increase $\mathbf{r}$, the smaller is our blind signature.

If we were to rely on an interactive argument, such as one-more-ISIS, we could prove security with only a few tweaks. Indeed, we would leverage the oracle to handle signing queries and would extract the one-more-ISIS solution from the forgery. However, as we restrict ourselves to standard assumptions we need to adapt our protocol. To handle signing queries, we will resort to tag-based pre-image sampling as in [MP12], which has a very important consequence for our construction. Indeed, instead of using the same matrix $\mathbf{A}$ for all issuances, we will use a matrix $\mathbf{A}_\mathfrak{t}$ which is parametrised by a tag $\mathfrak{t}$ which evolves across issuances. Fortunately, $\mathfrak{t}$ can be efficiently hidden in the zero-knowledge proof as illustrated in e.g. [dPLS18,LNPS21,JRS23,AGJ$^+$24] but as the commitment $\mathbf{c} = \mathbf{A}_\mathfrak{t}\mathbf{r}+\mathbf{d}\mathcal{H}(m)$ now depends on it, $\mathfrak{t}$ must be either transmitted by the signer to the user in a prior flow or selected by the user. We show that both options are possible but they lead to different variants with specific features. Concretely, the first option is conceptually the simplest one and yields the shortest blind signatures but it technically makes our protocol 3-round. While this is not ideal, we stress that this does not expose us to the ROS attack due to the very different structure of our protocol. Moreover, contrarily to the Schnorr-style approach, this extra round does not require to store secret values on the signer side in preparation for the third round (as the used tag $\mathfrak{t}$ can be made public), which facilitates state management. Nevertheless, we show for completeness in Section 6 that we can make our construction 2-round using the second approach which allows the user to select his own tag. This variant obviously removes this additional round but requires to implement safeguards to prevent a malicious user from selecting the same tag twice. For example, this can be done by maintaining a set of used tags on the signer side (and rejecting new requests associated with a tag in this list) or by deriving deterministically this tag using a hash function evaluated on some public information, when available. In all cases, it requires to work with a larger tag space (to avoid incidental collisions), which in turns affects the structure of $\mathbf{A}_{\mathfrak{t}}$ (to limit the reduction loss) and hence the system complexity.

Our blind signature then essentially consists in one vector $\mathbf{s}$ whose lower bits can be safely disclosed and whose higher bits are concealed in a zero-knowledge proof. Limiting to a minimum the size of the witness is the key to the performance of the system, leading to a blind signature size of only 40.47 KB when plugged in the framework by [AGJ+24] which provides all the necessary tools (tag-based pre-image sampling and ZK proofs) for our construction. In addition to achieving such compact sizes while being based on well-known lattice assumptions, our scheme is also reasonably efficient. We provide a proof-of-concept implementation in C[1] which showcases a concrete practicality. To our knowledge, this is the first implementation of lattice-based blind signatures. In particular, we achieve a reasonably fast issuance proof (the most computationally intensive part of issuance) of only 720ms with a not yet optimized implementation, which is orders of magnitude more efficient than when relying on general-purpose NIZKs as required by [BLNS23a].

2 Preliminaries

In this paper, for two integers $a \leq b$, we define $[a, b] = \{k \in \mathbb{Z} : a \leq k \leq b\}$. When $a = 1$, we simply use $[b]$ instead of $[1, b]$. Further, q is a positive integer, and we define $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. We may identify the latter with the set of representatives $(-q/2, q/2] \cap \mathbb{Z}$. Vectors are written in bold lowercase letters $\mathbf{a}$ and matrices in bold uppercase letters $\mathbf{A}$. The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^T$. The identity matrix of dimension d is denoted by $\mathbf{I}_d$. We use $\|\cdot\|_p$ to denote the ℓ_p norm of $\mathbb{R}^d$, i.e., $\|\mathbf{a}\|_p = (\sum_{i\in[d]} |a_i|^p)^{1/p}$ for any positive integer p, and $\|\mathbf{a}\|_\infty = \max_{i\in[d]} |a_i|$. We also define the spectral norm of a matrix $\mathbf{A}$ by $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq\mathbf{0}} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$.

2.1 Algebraic Number Theory

We now give the necessary notions in algebraic number theory. A more complete background can be found in Appendix A.1.1. We present our results over a power-of-two cyclotomic ring. We take n a power of two and let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ be the power-of-two cyclotomic ring of degree n. We also define $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ for any modulus $q \geq 2$. We sometimes use real-valued polynomials and consider elements in $K_\mathbb{R} = \mathbb{R}[x]/\langle x^n + 1\rangle$.

We use τ to denote the coefficient embedding, i.e., for all $a = \sum_{i\in[0,n-1]} a_i x^i \in R$, $\tau(a) = [a_0|\ldots|a_{n-1}]^T$. We sometimes use $\tau_i(a) = a_i$ to be the projection of $\tau(a)$ onto the i-th component. We also denote by M_τ the multiplication matrix map, defined by the relation $\tau(ab) = M_\tau(a)\tau(b)$. In power-of-two cyclotomic fields, this corresponds to the usual nega-circulant matrix featuring the coefficients of a. Then, the conjugate is defined by $a^* = a(x^{-1})$. These notations

[1] https://github.com/latticeblindsignature/lattice-blind-signature

are extended to vectors and matrices entry-wise, except that the conjugate of a matrix actually corresponds to the conjugate transpose. For an integer η, we define $S_\eta = \tau^{-1}([-\eta, \eta]^n)$, $\widetilde{S}_\eta = \tau^{-1}([-\eta, \eta-1]^n)$ and $T_\eta = \tau^{-1}([0, \eta]^n)$. We also define the usual vector norms $\|\cdot\|_p$ over R by $\|r\|_p := \|\tau(r)\|_p$, and the spectral norm $\|\mathbf{A}\|_2$ by $\|M_\tau(\mathbf{A})\|_2$.

2.2 Lattices

A full-rank *lattice* $\mathcal{L}$ of rank d is a discrete subgroup of $(\mathbb{R}^d, +)$. The *dual lattice* of $\mathcal{L}$ is defined by $\mathcal{L}^* = \{\mathbf{x} \in \mathrm{Span}_\mathbb{R}(\mathcal{L}) : \forall \mathbf{y} \in \mathcal{L}, \mathbf{x}^T\mathbf{y} \in \mathbb{Z}\}$. A lattice over R^d is identified with the lattice corresponding to its embedding into $\mathbb{R}^{nd}$. For any $\mathbf{A} \in R_q^{d\times m}$, we define the lattice $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in R^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod qR\}$, while for any $\mathbf{u} \in R_q^d$, we similarly define $\mathcal{L}_q^\mathbf{u}(\mathbf{A}) = \{\mathbf{x} \in R^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod qR\}$.

2.3 Probabilities

For a finite set S, we define $|S|$ to be its cardinality, and $U(S)$ to be the uniform probability distribution over S. We also let ψ_η be the centered binomial distribution of parameter $\eta \in \mathbb{N} \setminus \{0\}$ defined by the distribution of $\sum_{i\in[\eta]} a_i - b_i$ for $a_1, b_1, \ldots, a_\eta, b_\eta$ independently drawn from $U(\{0,1\})$. We then use $\mathcal{B}_\eta$ to denote the distribution over R whose coefficients follow ψ_η, i.e., $\tau^{-1}(\psi_\eta^n)$. We use $x \hookleftarrow \mathcal{P}$ to note the action of sampling $x \in S$ according to the probability distribution $\mathcal{P}$. In contrast, we use $x \sim \mathcal{P}$ when the random variable x follows $\mathcal{P}$.

Probabilistic Norm Bounds. The preimage sampler of [AGJ+24], which is an instantiation of [MP12], requires a bound a priori on the spectral norm of the secret trapdoor $\mathbf{R}$. We use the following heurisitically verified bound, used in a variety of works in the structured case, e.g. [MP12,GMPW20,LNP22]. Additionally, the security proof requires bounding projections $\mathbf{Rm}$. For that we use the following heuristic Johnson-Lindenstrauss-like bound which is empirically verified.

Lemma 2.1 ([AGJ+24, Lem. 2.2 & 2.4]). *Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power-of-two. Let d, m be two positive integers. It (heuristically) holds that $\mathbb{P}_{\mathbf{R}\sim\mathcal{B}_1^{d\times m}}[\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{d} + \sqrt{m} + 6)] = 1/O(1)$ (in particular non-negligible). Also, for an arbitrary $\mathbf{m} \in R^m$, it heuristically holds that $\mathbb{P}_{\mathbf{R}\sim\mathcal{B}_1^{d\times m}}[\|\mathbf{Rm}\|_2 \leq \frac{1}{\sqrt{2}}\sqrt{nd}\|\mathbf{m}\|_2] = 1/C$ with $C = O(1)$ (in particular non-negligible).*

Gaussian Measures. For a center $\mathbf{c} \in \mathbb{R}^d$ and positive definite $\mathbf{S} \in \mathbb{R}^{d\times d}$, we define the Gaussian function $\rho_{\sqrt{\mathbf{S}},\mathbf{c}} : \mathbf{x} \in \mathbb{R}^d \mapsto \exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c}))$. For a countable set $A \subseteq \mathbb{R}^d$, we define the *discrete Gaussian distribution* $\mathcal{D}_{A,\sqrt{\mathbf{S}},\mathbf{c}}$ of support A, covariance $\mathbf{S}$ and center $\mathbf{c}$ by its density $\mathcal{D}_{A,\sqrt{\mathbf{S}},\mathbf{c}} : \mathbf{x} \in A \mapsto \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x})/\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(A)$, where $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(A) = \sum_{\mathbf{x}\in A} \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x})$. When $\mathbf{c} = \mathbf{0}$, we omit it from the notations. When $\mathbf{S} = s^2\mathbf{I}_d$, we use s as subscript instead of $\sqrt{\mathbf{S}}$.

For $\mathbf{c} \in K_\mathbb{R}^d$ and a positive definite matrix $\mathbf{S} \in \mathbb{R}^{nd\times nd}$, we define the discrete Gaussian distribution over R^d by $\tau^{-1}(\mathcal{D}_{\tau(R^d),\sqrt{\mathbf{S}},\tau(\mathbf{c})})$, which we denote by

$\mathcal{D}_{R,\sqrt{\mathbf{S}},\mathbf{c}}$. Since $\tau(R^d) = \mathbb{Z}^{nd}$, the distribution corresponds to sampling an integer vector according to $\mathcal{D}_{\mathbb{Z}^{nd},\sqrt{\mathbf{S}},\tau(\mathbf{c})}$ which thus defines a vector of R^d via τ^{-1}. As coined by Micciancio and Regev [MR07], we define the *smoothing parameter* of a lattice $\mathcal{L}$, parameterized by $\varepsilon > 0$, by $\eta_\varepsilon(\mathcal{L}) = \inf\{s > 0 : \rho_{1/s}(\mathcal{L}^*) = 1 + \varepsilon\}$.

We first need a Gaussian regularity lemma from [GPV08, Lem. 5.2] generalized to non-spherical distributions. We state it over rings for coherence but it also applies to the integers.

Lemma 2.2 ([GPV08, Lem. 5.2] adapted). *Let d, m, q be positive integers, and $\overline{\mathbf{A}} \in R_q^{d\times m}$ such that $\overline{\mathbf{A}} R_q^m = R_q^d$. Then, let $\varepsilon \in (0, 1)$ and $\mathbf{S} \in \mathbb{R}^{nm\times nm}$ such that $\mathbf{S} - \eta_\varepsilon(\mathcal{L}_q^\perp(\overline{\mathbf{A}}))^2\mathbf{I}_{nm}$ is positive semi-definite. We finally define $\mathcal{P} = \overline{\mathbf{A}}\mathcal{D}_{R^m,\sqrt{\mathbf{S}}} \bmod qR$. It holds that $\forall \mathbf{x} \in R_q^d, \mathcal{P}(\mathbf{x}) \in [(1-\varepsilon)/(1+\varepsilon), 1+\varepsilon]q^{-nd}$.*

We now give the standard discrete Gaussian tail bound from [Ban93]. Notice that when $\mathbf{c} = \mathbf{0}$, the smoothing requirement $s \geq \eta_\varepsilon(\mathcal{L})$ in the following is not needed.

Lemma 2.3 ([Ban93, Lem. 1.5]). *Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d, and $s > 0$. Then, for all $c > 1/\sqrt{2\pi}$, we have*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{L},s}}\left[\|\mathbf{x}\|_2 > cs\sqrt{d}\right] < \left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^d.$$

We use c_d to denote the smallest $c > 1/\sqrt{2\pi}$ such that $(c\sqrt{2\pi e}e^{-\pi c^2})^d \leq 2^{-(\lambda+O(1))}$, where λ is the implicit security parameter.

Our scheme leverages the elliptic sampler $\mathsf{SamplePre}$ of [AGJ+24, Alg. 3.2] which corresponds to the sampler of [MP12] producing an elliptical distribution with two widths s_1, s_2 instead of a spherical one. Concretely, $\mathsf{SamplePre}$ takes a trapdoor $\mathbf{R}$, the matrix $\mathbf{A}'$ defining the first block $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$, a syndrome $\mathbf{u}$ to invert, a tag matrix $\mathbf{T} \in GL_d(R_q)$, and two Gaussian widths s_1, s_2 for the top and bottom parts. It then outputs a sample that is close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}([\mathbf{A}|\mathbf{TG}-\mathbf{AR}]),\sqrt{\mathbf{S}}}$ with $\mathbf{S} = \mathrm{diag}(s_1^2\mathbf{I}_{2d}, s_2^2\mathbf{I}_{kd})$. They provide a detailed security analysis and instantiation of the latter. The security proof of our blind signature relies in particular on the recent trapdoor switching argument proposed in [AGJ+24], which we recall here. It is based on the structure of the trapdoors of [MP12] and argues that one can extend the matrix to embed a partial trapdoor slot and change the sampling method unbeknownst to the adversary. This lemma is essential in the public key simulation step for the security of standard model signatures based on such trapdoors.

Lemma 2.4 ([AGJ+24, Lem. 4.1]). *Let d, q, k be positive integers, $b = \lceil q^{1/k}\rceil$. Let $\varepsilon \in (0, 1/4)$ and $s_{\mathbf{G}} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$. Then let $\mathbf{A}' \in R_q^{d\times d}$, $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$, $(\mathbf{R}_j)_{j\in[d+1]} \in (R^{2d\times k})^{d+1}$, and the partial gadget matrices $(\mathbf{G}_j)_{j\in[d]} = (\mathbf{e}_j \otimes [1|b|\ldots|b^{k-1}])_j \in (R^{d\times k})^d$. Let $(\mathfrak{t}_j)_{j\in[d+1]} \in (R_q^\times)^{d+1}$. Let $i \in [d]$. We define $\mathbf{G} = [\mathbf{G}_1|\ldots|\mathbf{G}_d]$, $\mathbf{R} = [\mathbf{R}_1|\ldots|\mathbf{R}_d]$ and $\mathbf{R}_{-i}$ the matrix where the block $\mathbf{R}_i$ in $\mathbf{R}$ has been replaced by $\mathbf{R}_{d+1}$. We also call $\mathbf{T} = \mathrm{diag}(\mathfrak{t}_1, \ldots, \mathfrak{t}_d)$ and $\mathbf{T}_{-i}$ the* *matrix* $\mathbf{T}$ *where the* i*-th diagonal entry is replaced by* $\mathfrak{t}_{d+1}$*. Let* s_1, s_2 *be two positive reals such that* $s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \max(\|\mathbf{R}\|_2, \|\mathbf{R}_{-i}\|_2)$ *and* $s_2 \geq \sqrt{2s_{\mathbf{G}}^2 + \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2}$*. Finally, fix* $\mathbf{u} \in R_q^d$*.*

We call $\overline{\mathbf{A}}$ *the matrix* $[\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1}] \bmod qR$ *for clarity, and then define the following distributions.*

$\mathcal{P}_1$ *Sample* $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$*,* $(\mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1})\mathbf{v}_3 \bmod qR, \mathbf{T}, s_1, s_2)$ *and output* $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$*.*

$\mathcal{P}_2$ *Sample* $\mathbf{v}_{2,i} \hookleftarrow \mathcal{D}_{R^k, s_2}$*,* $(\mathbf{v}_1, (\mathbf{v}_{2,1}, \ldots, \mathbf{v}_{2,i-1}, \mathbf{v}_3, \mathbf{v}_{2,i+1}, \ldots, \mathbf{v}_{2,d})) \leftarrow \mathsf{SamplePre}(\mathbf{R}_{-i}, \mathbf{A}', \mathbf{u} - (\mathfrak{t}_i\mathbf{G}_i - \mathbf{AR}_i)\mathbf{v}_{2,i} \bmod qR, \mathbf{T}_{-i}, s_1, s_2)$*, define and output* $(\mathbf{v}_1, (\mathbf{v}_{2,j})_{j\in[d]}, \mathbf{v}_3)$*.*

It holds that $\forall \mathbf{v} \in \mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}), \mathcal{P}_1(\mathbf{v}) \in [\delta^{-1}, \delta] \cdot \mathcal{P}_2(\mathbf{v})$*, where*

$$\delta = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{12d(n-1)+5} \left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{2ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2(12d(n-1)+7)\varepsilon$$

Rejection Sampling. Our security proof relies on a rejection sampling argument proven in [LNS21].

Lemma 2.5 ([LNS21, Lem. 3.2]). *Let* d *be a positive integer, and* $S \subset R^d$ *a set of vectors of Euclidean norm at most* $T > 0$*. Let* $\mathscr{D}_S$ *be a distribution over* S*. Let* $M > 1$*,* $\alpha = \sqrt{\pi / \ln M}$ *and* $s \geq \alpha T$*. We then define the following distributions.*

$\mathcal{P}_1$ *Sample* $\mathbf{s} \hookleftarrow \mathscr{D}_S$*,* $\mathbf{y} \hookleftarrow \mathcal{D}_{R^d,s}$ *and set* $\mathbf{z} = \mathbf{y} + \mathbf{v}$*. Then, sample* $u \hookleftarrow U([0,1))$*. If* $\langle \tau(\mathbf{z}), \tau(\mathbf{s})\rangle < 0$ *or if* $u > \frac{1}{M}\exp(\pi(\|\tau(\mathbf{s})\|_2^2 - 2\langle\tau(\mathbf{z}), \tau(\mathbf{s})\rangle)/s^2)$*, output* $\perp$*. Otherwise, output* $(\mathbf{s}, \mathbf{z})$*.*

$\mathcal{P}_2$ *Sample* $\mathbf{s} \hookleftarrow \mathscr{D}_S$*, and* $\mathbf{z} \hookleftarrow \mathcal{D}_{R^d,s}$*. Then, sample* $u \hookleftarrow U([0,1))$*. If* $\langle\tau(\mathbf{z}), \tau(\mathbf{s})\rangle < 0$ *or if* $u > \frac{1}{M}$*, output* $\perp$*. Otherwise, output* $(\mathbf{s}, \mathbf{z})$*.*

Conditioned on not aborting, it holds that $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *are identical.*

2.4 High and Low Decomposition

Our blind signature scheme relies on decomposing some vectors into low order bits and high order bits, so as to only hide the high part. For that, we use the decomposition functions from [CCD+23] but tweaked to meet the requirements of our system. We consider b to be a power of two.

Algorithm 2.1: High(x, b)

return $2\lfloor x/2b \rfloor + 1$

Algorithm 2.2: Low(x, b)

return $x - b \cdot \mathsf{High}(x, b)$

Algorithm 2.3: Decompose(x, b)

return $(x_H, x_L) = (\mathsf{High}(x, b), \mathsf{Low}(x, b))$ ▷ Verifies $x_L + bx_H = x$

To prove the security of our scheme, we need the following two lemmas related to the low and high order distribution of a uniform variable.

Lemma 2.6. *Let b be a power of two. It holds that $\mathsf{Low}(U([-2b, 2b-1]), b) = U([-b, b-1])$ and $\mathsf{High}(U([-2b, 2b-1]), b) = U(\{-1, 1\})$. Additionally, it holds that $U([-b, b-1]) + b \cdot U(\{-1, 1\}) = U([-2b, 2b-1])$.*

Proof. Let x be in $[-2b, 2b-1]$. If $x \in [-2b, -1]$, then we have $\mathsf{High}(x, b) = -1$ which implies that $\mathsf{Low}(x, b) = x + b$. If $x \in [0, 2b-1]$, then $\mathsf{High}(x, b) = 1$ which implies that $\mathsf{Low}(x, b) = x - b$.

We thus directly get that $\mathsf{High}(U([-2b, 2b-1]), b) = U(\{-1, 1\})$ as both 1 and -1 happen with probability $2b/4b = 1/2$. Also, from the expressions of $\mathsf{Low}(\cdot, b)$ on each interval, we deduce that for $x' \in [-b, b-1]$ it holds

$$\begin{aligned}
\mathbb{P}_{x \sim U([-2b,2b-1])}[\mathsf{Low}(x, b) = x'] &= \mathbb{P}[x + b = x' \wedge x < 0] + \mathbb{P}[x - b = x' \wedge x \geq 0] \\
&= \mathbb{P}[x = x' - b] + \mathbb{P}[x = x' + b] \\
&= 1/2b
\end{aligned}$$

As a result $\mathsf{Low}(U([-2b, 2b-1]), b) = U([-b, b-1])$. Next, consider $x \sim U([-b, b-1])$, $y \sim U(\{-1, 1\})$ and define $z = x + by$. Then z takes values in $[-2b, 2b-1]$ and for all possible outcome z', we have

$$\begin{aligned}
\mathbb{P}_z[z = z'] &= \frac{1}{2}\mathbb{P}_x[x = z' + b] + \frac{1}{2}\mathbb{P}_x[x = z' - b] \\
&= \frac{1}{4b}(\mathbf{1}(z' + b \in [-b, b-1]) + \mathbf{1}(z' - b \in [-b, b-1])) \\
&= \frac{1}{4b},
\end{aligned}$$

where the last equality comes from the fact that $z' + b$ and $z' - b$ are distant by $2b$ and thus cannot be in $[-b, b-1]$ simultaneously, but because $z' \in [-2b, 2b-1]$ at least one of them is. □

The motivation for this decomposition is to (1) minimize the size of x, while (2) still having independent low and high part when x is uniform, and (3) still have enough entropy in the high bits. The latter two points will be essential for proving security of our construction. Another property we need from our decomposition is the invariance of the uniform distribution by shift, which we state in the following.

Lemma 2.7. *Let b be a power of two. It holds that for any $y \in \mathbb{Z}$, $\mathsf{Low}(y + U([-b, b-1]), b) = U([-b, b-1])$.*

Proof. We define $y_H = \mathsf{High}(y, b)$ and $y_L = \mathsf{Low}(y, b)$. For any $x \in [-b, b-1]$, we have that $\lfloor y/2b \rfloor - 1/2 \leq (x + y)/2b \leq \lfloor y/2b \rfloor + 3/2$. As a result, we have $\mathsf{High}(x + y, b) \in \{y_H - 2, y_H, y_H + 2\}$ and thus $\mathsf{Low}(x + y, b) \in \{x + y_L + 2b, x + y_L, x + y_L - 2b\}$. Hence, for $x' \in [-b, b-1]$, it holds that

$$\mathbb{P}_{x \sim U([-b,b-1])}[\mathsf{Low}(x + y, b) = x'] = \mathbb{P}_{x \sim U([-b,b-1])}[x \in S]],$$

with $S = \{x' - y_L - 2b, x' - y_L, x' - y_L + 2b\} \cap [-b, b-1]$. Because $x', y_L \in [-b, b-1]$, we always have $|S| = 1$. Indeed, $-2b + 1 \leq x' - y_L \leq 2b - 1$. So if we assume towards contradiction that $S = \varnothing$, it directly means that either $-2b+1 \leq x' - y_L \leq -b-1$ in which case $x' - y_L + 2b \in S$, or $b \leq x' - y_L \leq 2b-1$ in which case $x' - y_L - 2b \in S$. Either way, we get a contradiction proving that $|S| \geq 1$. Then, if we assume that S contains two distinct elements $A \neq B$, it holds that $|A - B| \geq 2b$. As a result, both A and B cannot be in S simultaneously, which proves that $|S| \leq 1$. In the end, because $|S| = 1$, it holds that

$$\mathbb{P}_{x \sim U([-b,b-1])}[\mathsf{Low}(x+y,b) = x'] = \frac{1}{2b},$$

thus proving that $\mathsf{Low}(y + U([-b, b-1]), b) = U([-b, b-1])$. $\square$

2.5 Hardness Assumptions

The security of our blind signature is based on the *Module Learning With Errors* (M-LWE) and *Module Short Integer Solution* (M-SIS) problems [LS15]. We consider both problems in their Hermite Normal Form, i.e., we use the same distribution for the M-LWE secret and error, and we specify the identity in the M-SIS matrix.

Definition 2.1 (M-LWE). *Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power-of-two. Let d, m, k, q be positive integers and $\mathscr{D}_r$ a distribution on R. The Module Learning With Errors problem $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$ asks to distinguish between the following distributions: (1) $(\mathbf{A}', [\mathbf{I}_m|\mathbf{A}']\mathbf{R} \bmod qR)$, where $\mathbf{A}' \sim U(R_q^{m\times d})$ and $\mathbf{R} \sim \mathscr{D}_r^{d+m\times k}$, and (2) $(\mathbf{A}', \mathbf{B})$, where $\mathbf{A}' \sim U(R_q^{m\times d})$ and $\mathbf{B} \sim U(R_q^{m\times k})$.*

The advantage of a PPT adversary $\mathcal{A}$ against $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$ is

$$\text{Adv}_{\text{M-LWE}}[\mathcal{A}] = |\mathbb{P}[\mathcal{A}(\mathbf{A}', [\mathbf{I}_m|\mathbf{A}']\mathbf{R}) = 1] - \mathbb{P}[\mathcal{A}(\mathbf{A}', \mathbf{B}) = 1]|,$$

When the parameters are clear from the context, we define the hardness bound as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{A}\ \mathsf{PPT}} \text{Adv}_{\text{M-LWE}}[\mathcal{A}]$. Additionally, a standard hybrid argument shows that $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$ is at least as hard as $\text{M-LWE}^1_{n,d,m,q,\mathscr{D}_r}$ at the expense of a loss factor k in the reduction.

Definition 2.2 (M-SIS). *Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power-of-two. Let d, m, q be positive integers and $\beta > 0$ with $m > d$. The Module Short Integer Solution problem in Hermite Normal Form $\text{M-SIS}_{n,d,m,q,\beta}$ asks to find $\mathbf{x} \in \mathcal{L}_q^\perp([\mathbf{I}_d|\mathbf{A}']) \setminus \{\mathbf{0}\}$ such that $\|\mathbf{x}\|_2 \leq \beta$, given $\mathbf{A}' \hookleftarrow U(R_q^{d\times m-d})$.*

The advantage of a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ against $\text{M-SIS}_{n,d,m,q,\beta}$ is defined by

$$\text{Adv}_{\text{M-SIS}}[\mathcal{A}] = \mathbb{P}\left[[\mathbf{I}_d|\mathbf{A}']\mathbf{x} = \mathbf{0} \bmod qR \wedge 0 < \|\mathbf{x}\|_2 \leq \beta : \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}')\right],$$

where the probability is over the randomness of $\mathbf{A}'$ and the random coins of $\mathcal{A}$. When the parameters are clear from the context, we define the hardness bound as $\varepsilon_{\text{M-SIS}} = \sup_{\mathcal{A}\ \mathsf{PPT}} \text{Adv}_{\text{M-SIS}}[\mathcal{A}]$. We now present the M-LWE problem in its multiple secrets variant which we use throughout the paper.

2.6 Blind Signatures

Blind signatures extend the standard notion of digital signatures by requiring a blindness property which prevents the signer from tracing a presented signature to a specific issuance. This blindness logically affects the signing process as the signed message must be concealed but also the unforgeability notion as the signer can only keep track of the number of signed messages, not their actual content.

We use the definition and security properties from [JLO97] that we recall below. A blind signature is a collection of three algorithms Setup, KeyGen, Verify and an interactive protocol Sign between a signer and a user. The Setup algorithms takes as input a security parameter λ and outputs the public parameters of the system pp. Then, the KeyGen algorithms uses pp and generates a pair of keys $(\mathsf{pk}, \mathsf{sk})$ for the signer. A user can then interact in Sign with a signer holding sk to sign a message m. In the end, the user gets back a signature sig and the signer gets no output. Finally, the Verify algorithm takes as input pp, pk as well as the message-signature pair (m, sig) and outputs 1 if the blind signature is valid, and 0 otherwise.

From the security perspective, we expect a blind signature scheme to be *correct*, *one-more unforgeable* and *blind*. We define these notions precisely here in accordance with [JLO97]. We note that in the blindness game, the adversary may choose the public parameters and keys maliciously.

Definition 2.3 (Correctness). *A blind signature* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is correct if for any* λ, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, *message* m *and signature* sig *outputted to* U *in the execution of* $\mathsf{Sign}\langle S(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}), U(\mathsf{pp}, \mathsf{pk}, m)\rangle$, *it holds that* $\mathsf{Verify}(\mathsf{pk}, m, \mathsf{sig}, \mathsf{pp}) = 1$ *with probability* $1 - \mathsf{negl}(\lambda)$.

Definition 2.4 (One-More Unforgeability). *Let* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *define a blind signature. We define the following game.*

Challenger $\mathcal{C}$		***Adversary*** $\mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$	$\xrightarrow{\mathsf{pp}, \mathsf{pk}}$	
	Blind Signature Queries *(at most* Q*)*	
$(\perp, \mathsf{sig}_i) \leftarrow$	$\mathsf{Sign}\langle \mathcal{C}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}), \mathcal{A}(\mathsf{pp}, \mathsf{pk}, m_i)\rangle$ $\longleftrightarrow$	*Choose* m_i
	$\xleftarrow{(m_i^\star, \mathsf{sig}_i^\star)_{i\in[Q+1]}}$	*Choose* $(m_i^\star, \mathsf{sig}_i^\star)_{i\in[Q+1]}$
$\mathcal{A}$ *wins if* $\forall i \neq j, m_i^\star \neq m_j^\star$ *and* $\forall i, \mathsf{Verify}(\mathsf{pk}, m_i^\star, \mathsf{sig}_i^\star, \mathsf{pp}) = 1$		

The advantage of the adversary $\mathcal{A}$ *is its probability of winning the above game, that is*

$$Adv_{OM\text{-}UF}[\mathcal{A}] = \mathbb{P}[\forall i \in [Q+1], \mathsf{Verify}(\mathsf{pk}, m_i^\star, \mathsf{sig}_i^\star, \mathsf{pp}) = 1 \wedge \forall i \neq j, m_i^\star \neq m_j^\star]$$

We say that the blind signature is one-more unforgeable if the advantage of any probabilistic polynomial-time adversary $\mathcal{A}$ is negligible in λ.

Definition 2.5 (Blindness). *Let* (Setup, KeyGen, Sign, Verify) *define a blind signature scheme. We define the following game.*

Challenger $\mathcal{C}$		***Adversary*** $\mathcal{A}$
	$\xleftarrow{\mathsf{pp}, \mathsf{pk}, m_0, m_1}$	*Choose* $\mathsf{pp}, \mathsf{pk}, \mathsf{sk}, m_0, m_1$
$\rho \hookleftarrow U(\{0,1\})$		
	$(\perp, \mathsf{sig}_0) \leftarrow \mathsf{Sign}\langle \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}), \mathcal{C}(\mathsf{pp}, \mathsf{pk}, m_\rho)\rangle$ $\longleftrightarrow$	
	$(\perp, \mathsf{sig}_1) \leftarrow \mathsf{Sign}\langle \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}), \mathcal{C}(\mathsf{pp}, \mathsf{pk}, m_{1-\rho})\rangle$ $\longleftrightarrow$	
	$\xrightarrow{\mathsf{sig}_0, \mathsf{sig}_1}$	
	$\xleftarrow{\rho^\star}$	*Choose* $\rho^\star \in \{0,1\}$
$\mathcal{A}$ *wins if* $\rho^\star = \rho$		

The advantage of the adversary $\mathcal{A}$ is $Adv_{Blind}[\mathcal{A}] = |\mathbb{P}[\rho^\star = \rho] - 1/2|$. We say that the blind signature satisfies blindness if the advantage of any probabilistic polynomial-time adversary $\mathcal{A}$ is negligible in λ.

3 Blind Signature from Lattices

We now introduce our new blind signature scheme. It requires a variety of tools that we choose so as to leverage synergies. More precisely, we need a commitment, a public key encryption, a signature that interfaces to sign commitments, and zero-knowledge proof systems. Most of these tools are provided in the recent framework of *signatures with efficient protocols* from [AGJ+24], which internally uses a concrete instantiation of the Micciancio-Peikert sampler [MP12], and the zero-knowledge proof system of [LNP22]. Our scheme also embarks the proof system optimizations from [LNP22, Sec. 4.4, App. A] and [LN22, Sec. 3]. Although the authors of [AGJ+24] use a simple Ajtai commitment from part of the signature matrix, which could be used in our case as well, we aim at salvaging the commitment randomness to blind part of the signature to minimize the amount of secret data in the zero-knowledge proof. As mentioned in the introduction, we thus use the full signature matrix $\mathbf{A}_\mathfrak{t} = [\mathbf{A}|\mathfrak{t}\mathbf{G} - \mathbf{B}|\mathbf{A}_3]$ for the commitment. Finally, the encryption mechanism only has a detection role in most blind signature constructions. As such, we do not need it to interact with the signature per se, but the encryption circuit must be provable in zero-knowledge, while minimizing the witness size to keep proofs short. For that we use a module version of Regev's encryption scheme [Reg09].

3.1 The Construction

Algorithm 3.1: BS.Setup

Input: Security parameter λ.

1. Choose a positive integer d.
2. Choose an odd prime q s.t. $q = 5 \bmod 8$.
3. Choose positive integer w. ▷ Hamming weight of tags
4. Choose positive integer b. ▷ Gadget base
5. Choose positive integers b_1, b_2. ▷ Decomposition bases
6. $\mathcal{T}_w \leftarrow \{\mathsf{t} \in T_1 : \|\mathsf{t}\|_1 = w\}$. ▷ Tag space
7. $k \leftarrow \lceil \log_b q \rceil$.
8. $\mathbf{G} = \mathbf{I}_d \otimes [1 \cdots b^{k-1}] \in R_q^{d \times dk}$. ▷ Gadget matrix
9. $r \leftarrow \sqrt{\ln(2nd(2+k)(1+\varepsilon^{-1}))/\pi}$. ▷ $r \gtrsim \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})$
10. $s_{\mathbf{G}} \leftarrow r\sqrt{b^2+1}$. ▷ Gadget sampling width
11. $s_1 \leftarrow \max\left(\sqrt{\frac{\pi}{\ln(2)}}(n\sqrt{d} + 2b_1\sqrt{2nd}), \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2 - 1}} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)\right)$. ▷ Top preimage width
12. $s_2 \leftarrow \max(r\sqrt{2b^2+3}, \sqrt{\frac{\pi}{\ln(2)}} b_2 \sqrt{nk(d+1)})$. ▷ Bottom preimage width
13. $\alpha_1 \leftarrow \frac{s_1}{n\sqrt{d} + 2b_1\sqrt{2nd}}$ ▷ Rejection sampling slack
14. $\alpha_2 \leftarrow \frac{s_2}{b_2\sqrt{nk(d+1)}}$
15. $M_i \leftarrow \exp(\pi/\alpha_i^2)$ for $i \in [2]$.
16. $\mathsf{seed} \hookleftarrow U(\{0,1\}^{256})$
17. $\mathbf{d} \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_1)$. ▷ follows $U(R_q^d)$
18. $\mathbf{A}' \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_2)$. ▷ follows $U(R_q^{d \times d})$
19. $\mathbf{A}_3 \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_3)$. ▷ follows $U(R_q^{d \times k})$
20. $\mathbf{u} \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_4)$. ▷ follows $U(R_q^d)$
21. $\mathbf{A} \leftarrow [\mathbf{I}_d | \mathbf{A}'] \in R_q^{d \times 2d}$.

    Verifiable Encryption.
22. Choose a prime $p \neq q$ ▷ Verifiable encryption modulus
23. Choose positive integers m_e, d_e, η_e ▷ Verifiable encryption parameters
24. $\mathbf{A}_e \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_5)$. ▷ follows $U(R_p^{m_e \times d_e})$
25. $\mathbf{b}_e \leftarrow \mathcal{H}(\mathsf{seed}, \mathsf{domain\_separator}_6)$. ▷ follows $U(R_p^{m_e})$

Output: $\mathsf{pp} = (\lambda, n, d, q, w, b, k, r, s_{\mathbf{G}}, s_1, s_2, \alpha_1, \alpha_2, M_1, M_2, \mathsf{seed})$.

Algorithm 3.2: BS.KeyGen

Input: Public parameters pp as in Algorithm 3.1.

1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d \times dk}$ conditioned on $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$.
2. $\mathbf{B} \leftarrow \mathbf{AR} \bmod qR \in R_q^{d \times dk}$.

Output: $\mathsf{pk} = \mathbf{B}$, and $\mathsf{sk} = \mathbf{R}$.

Algorithm 3.3: BS.Sign

Input: Signer S with $\mathsf{pp}, \mathsf{pk}, \mathsf{sk}$, and user U with pp, pk and message $m \in \{0,1\}^*$.

Signer S.
1. $\mathsf{t} \leftarrow F(\mathsf{st})$.

2. $\mathsf{st} \leftarrow \mathsf{st} + 1$
3. Send t to U.

<u>User U.</u>

4. **if** $\mathsf{t} \notin \mathcal{T}_w$, **abort**.
5. $\mathbf{r}_{1,L} \hookleftarrow U(\widetilde{S}_{b_1}^{2d})$, $\mathbf{r}_{1,H} \hookleftarrow \tau^{-1}(U(\{-1,1\}^{2nd}))$.
6. $\mathbf{r}_1 \leftarrow \mathbf{r}_{1,L} + b_1 \mathbf{r}_{1,H}$.
7. $\mathbf{r}_2 \hookleftarrow U(\widetilde{S}_{b_2}^{kd})$, $\mathbf{r}_3 \hookleftarrow U(\widetilde{S}_{b_2}^{k})$.
8. $\mathbf{c} \leftarrow \mathbf{A}\mathbf{r}_1 + (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{r}_2 + \mathbf{A}_3\mathbf{r}_3 + \mathbf{d}\mathcal{H}(m) \bmod qR$. $\triangleright$ $\mathcal{H}(m) \in T_1$
9. $\mathbf{r}_e \hookleftarrow \mathcal{B}_{\eta_e}^{m_e}$.
10. $\mathsf{ct}_0 \leftarrow \mathbf{A}_e^T \mathbf{r}_e \bmod pR$.
11. $\mathsf{ct}_1 \leftarrow \mathbf{b}_e^T \mathbf{r}_e + \lfloor p/2 \rceil \mathcal{H}(m) \bmod pR$.
12. $\pi_1 \leftarrow \mathsf{Prove}_1(\mathbf{r}_1, [\mathbf{r}_2|\mathbf{r}_3], \mathbf{r}_e, \mathcal{H}(m))$.
13. Send $(\mathbf{c}, \mathsf{ct}_0, \mathsf{ct}_1, \pi_1)$ to S.

<u>Signer S.</u>

14. **if** $\mathsf{Verify}_1(\pi_1) = 0$, **abort**.
15. $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
16. $\mathbf{v} = [\mathbf{v}_1^T | \mathbf{v}_2^T]^T \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{c} - \mathbf{A}_3\mathbf{v}_3, \mathsf{t}, s_1, s_2)$.
17. Parse $\mathbf{v}_1$ into $[\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$ with $\mathbf{v}_{1,1}, \mathbf{v}_{1,2} \in R^d$
18. Send $(\mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ to U.

<u>User U.</u>

19. $\mathbf{v}_{1,1} \leftarrow \mathbf{u} + \mathbf{c} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR$.
20. $\rho \leftarrow (\|[\mathbf{v}_{1,1}|\mathbf{v}_{1,2}]\|_2^2 \le B_1^2) \wedge (\|[\mathbf{v}_2|\mathbf{v}_3]\|_2^2 \le B_2^2)$.
21. **if** $\rho \neq 1$, **abort**.
22. $(\mathbf{w}_{1,H}, \mathbf{w}_{1,L}) \leftarrow \mathsf{Decompose}(\mathbf{v}_1 - \mathbf{r}_{1,L}, b_1) - (\mathbf{r}_{1,H}, \mathbf{0})$
23. $(\mathbf{w}_{i,H}, \mathbf{w}_{i,L}) \leftarrow \mathsf{Decompose}(\mathbf{v}_i - \mathbf{r}_i, b_2)$ for $i \in \{2,3\}$.
24. $\pi_2 \leftarrow \mathsf{Prove}_2(\mathsf{t}, \mathbf{w}_{1,H}, \mathbf{w}_{2,H}, \mathbf{w}_{3,H})$.

Output: S gets $\perp$, and U gets $\mathsf{sig} = (\mathbf{w}_{1,L}, \mathbf{w}_{2,L}, \mathbf{w}_{3,L}, \pi_2)$.

Algorithm 3.4: BS.Verify

Input: Public key pk, message m, blind signature $\mathsf{sig} = (\mathbf{w}_{1,L}, \mathbf{w}_{2,L}, \mathbf{w}_{3,L}, \pi_2)$, and public parameters pp.
Output: $\mathsf{Verify}_2(\pi_2) \wedge (\mathbf{w}_{1,L} \in \widetilde{S}_{b_1}^{2d}) \wedge ([\mathbf{w}_{2,L}|\mathbf{w}_{3,L}] \in \widetilde{S}_{b_2}^{k(d+1)}) \wedge (\mathcal{H}(m) \in T_1)$.

To remain as general as possible, our protocol description does not make any assumption on the message m which is simply parsed as a bitstring. The mapping to the appropriate set is then done through a hash function $\mathcal{H}$ which only add the mild requirement that the produced digests do not collide. We stress that, in the blind issuance process, π_1 is a proof of knowledge of $\mathcal{H}(m)$, and not of m. In particular, we do *not* need to prove that $\mathcal{H}(m)$ is indeed the output of some hash function, which is key to the performance of our scheme.

Actually, we treat $\mathcal{H}$ as an extended output function such as SHAKE256, which is why we use the same notation all along our protocol although the output space depends on which element is expanded.

3.2 Verifiable Encryption

The verifiable encryption is there to detect, in the one-more unforgeability proof, the actual forgery so as to extract the witnesses of only one zero-knowledge proof. Concretely, the challenger will use its knowledge of the decryption keys to keep track of each message it has signed and thus be able to identify the one(s) it has not. This naturally conflicts with the blindness property where the signer is now controlled by the adversary and thus cannot access to such a knowledge. As usual, this problem is dealt with by resorting to the random oracle model which enables to set the public key $\mathbf{b}_e$ either as a uniformly random element (standard use and blindness proof) or as $\mathbf{A}_e\mathbf{s}_e + \mathbf{e}_e \bmod pR$ for $\mathbf{A}_e \hookleftarrow U(R_p^{m_e \times d_e})$ and $(\mathbf{s}_e, \mathbf{e}_e) \hookleftarrow \mathcal{B}_{\eta_e}^{d_e+m_e}$ (one-more unforgeability proof). In the latter case $(\mathbf{s}_e, \mathbf{e}_e)$ serves as decryption key and uniformity of $\mathbf{b}_e$ is argued under M-LWE. A zero-knowledge proof ensures well-formedness of the ciphertexts (that may be generated by the adversary) but note that it only enforces a certain Euclidean bound $B_{r,e}$ on $\mathbf{r}_e^\star$, not that it follows the expected distribution. We thus need to set the parameters so that $\left\|\mathbf{e}_e^T\mathbf{r}_e^\star\right\|_\infty < p/2$ based on this sole requirement. For that, we rely on $\left\|\mathbf{e}_e^T\mathbf{r}_e^\star\right\|_\infty \leq \|\mathbf{e}_e\|_2 \cdot \|\mathbf{r}_e^\star\|_2$ and use a concentration bound for the binomial distribution to upper-bound $\|\mathbf{e}_e\|_2$ (which is selected by the challenger and thus follows the prescribed distribution).

Binomial Tail Bound. We can obtain a tail bound for the binomial distribution using Hoeffding's inequality on $X = \|\mathbf{x}\|_2^2$. In particular, X is a sum of i.i.d random variables which all take values in $[0, \eta_e^2]$. Additionally, the expectation of X is exactly $\eta_e/2 \cdot N$ where $N = nm_e$ is the dimension of $\tau(\mathbf{x})$. As a result, Hoeffding's inequality shows that

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{B}_{\eta_e}^{m_e}}[\|\mathbf{x}\|_2^2 - nm_e\eta_e/2 > t] \leq \exp(-2t^2/nm_e\eta_e^4),$$

which can be written as

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{B}_{\eta_e}^{m_e}}\left[\|\mathbf{x}\|_2 > \sqrt{\frac{\eta_e nm_e}{2}}\sqrt{1+\eta_e\sqrt{\frac{2\lambda}{nm_e\log_2 e}}}\right] \leq 2^{-\lambda}.$$

For typical parameters (e.g., $n = 256$, $m_e = 7$, $\lambda = 128$, $\eta_e = 1$) we get a bound of around $B_{r,e} = t\sqrt{\eta_e nm_e/2}$ for $t \approx 1.15$. When η_e is small, we can actually derive the exact probability. We do it for the case $\eta_e = 1$ which is used in our scheme. We first observe that for each coefficient of $\tau(\mathbf{x})$, $\tau_i(\mathbf{x}) \sim \psi_1$ and therefore $\tau_i(\mathbf{x})^2 \sim U(\{0,1\})$. We now consider the polynomial $P = 1/2 + 1/2 \cdot T \in \mathbb{Q}[T]$, and define $P_X = P^N = P^{nm_e}$. We can express P_X directly as $P_X = 2^{-nm_e}\sum_{i=0}^{nm_e}\binom{nm_e}{i}T^i$. Finally, we observe that the probability that X equals i is exactly the i-th coefficient of P_X, that is $\binom{nm_e}{i}/2^{nm_e}$. For each $\alpha \in [0, nm_e]$, we can then compute exactly

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{B}_1^{m_e}}\left[\|\mathbf{x}\|_2^2 > \alpha\right] = 1 - \sum_{i=0}^{\alpha} 2^{-nm_e}\binom{nm_e}{i}.$$

For $n = 256, m_e = 7, \lambda = 128$, we find that the probability is less than $2^{-\lambda}$ for $\alpha = 1170$ which corresponds to a tail cut $t \approx 1.14272$. As it only affects the verifiable encryption which represents only a small fraction of the issuance phase, we use the tailcut $t = 1.15$ and thus the proven bound $B_{r,e} = t\sqrt{\eta_e n m_e/2}$, giving us a security margin on the tail bound.

Security. In the security proofs of our blind signature, we need both the fact that ciphertexts are indistinguishable from uniform, and that the public key (when embedding a secret key in the one-more unforgeability proof) is indistinguishable from uniform as well. The latter simply relies on the M-LWE$_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ assumption. The former relies on the fact that for perfectly uniform $(\mathbf{A}_e, \mathbf{b}_e)$, the vector

$$\begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e \bmod pR$$

is uniform. This is argued using the Knapsack formulation of M-LWE which here reduces from M-LWE$_{n,m_e-(d_e+1),m_e,p,\mathcal{B}_{\eta_e}}$. Choosing $m_e = 2d_e + 1$ thus means that both arguments rely on the same M-LWE assumption.

3.3 Relations of Zero-Knowledge Arguments

Our blind signature requires two zero-knowledge arguments. The first one, employed in the issuance phase, aims at proving the commitment and the ciphertext are well-formed by proving knowledge of $(\mathbf{r}_1, \mathbf{r}_{23}, \mathbf{r}_e, h) \in R^{2d+k(d+1)+m_e+1}$ such that

$$\mathbf{A}\mathbf{r}_1 + [\mathsf{t}\mathbf{G} - \mathbf{B}|\mathbf{A}_3]\mathbf{r}_{23} + \mathbf{d}h = \mathbf{c} \bmod qR, \quad \|\mathbf{r}_1\|_2 \le B_{r,1}, \|\mathbf{r}_{23}\|_2 \le B_{r,2}, \quad h \in T_1$$
$$\mathbf{A}_e^T\mathbf{r}_e = \mathsf{ct}_0 \bmod pR, \quad \mathbf{b}_e^T\mathbf{r}_e + \lfloor p/2 \rceil h = \mathsf{ct}_1 \bmod pR, \quad \text{and} \quad \|\mathbf{r}_e\|_2 \le B_{r,e},$$

where $B_{r,1} = 2b_1\sqrt{2nd}$, $B_{r,2} = b_2\sqrt{nk(d+1)}$, and $B_{r,e} = t\sqrt{\eta_e n m_e/2}$ and $\mathbf{r}_{23} = [\mathbf{r}_2^T|\mathbf{r}_3^T]^T$. We recall here that there is no need to prove correct hash evaluation as the witness is the digest $h = \mathcal{H}(m)$ of the message m and not m itself. In particular, our security proofs will not make any assumption on h beyond that it belongs to the set T_1.

The second argument is used to finalize the blind signature and proves knowledge of $(\mathbf{w}_{1,H}, \mathbf{w}_{2,H}, \mathbf{w}_{3,H}, \mathsf{t}) \in R^{2d+k(d+1)+1}$ such that

$$\begin{aligned} b_1\mathbf{A}\mathbf{w}_{1,H} + \mathsf{t}\mathbf{G}\mathbf{w}_{2,L} + b_2\mathsf{t}\mathbf{G}\mathbf{w}_{2,H} - b_2\mathbf{B}\mathbf{w}_{2,H} + b_3\mathbf{A}_3\mathbf{w}_{3,H} \\ = \mathbf{u} + \mathbf{d}\cdot\mathcal{H}(m) - \mathbf{A}\mathbf{w}_{1,L} + \mathbf{B}\mathbf{w}_{2,L} - \mathbf{A}_3\mathbf{w}_{3,L} \bmod qR, \\ \|\mathbf{w}_{1,H}\|_2^2 \le {B'_1}^2, \quad \|[\mathbf{w}_{2,H}|\mathbf{w}_{3,H}]\|_2^2 \le {B'_2}^2, \quad \text{and} \quad \mathsf{t} \in \mathcal{T}_w, \end{aligned}$$

where $(\mathbf{w}_{1,L}, \mathbf{w}_{2,L}, \mathbf{w}_{3,L})$ is included in the blind signature, and $B'_1 = \lfloor (B_1/b_1 + 3\sqrt{2nd})^2 \rfloor^{1/2}$, and $B'_2 = \lfloor (B_2/b_2 + 2\sqrt{nk(d+1)})^2 \rfloor^{1/2}$.

The full description of these arguments is deferred to Sections A.3 and A.4 respectively.

4 Security Analysis

We now prove the correctness, one-more unforgeability and blindness of our scheme.

4.1 Correctness

Lemma 4.1. *The blind signature of Algorithms 3.1, 3.2, 3.3 and 3.4 is correct.*

Proof. First, we study the possible aborts of a legitimate signature generation. Since F maps to $\mathcal{T}_w$, the user does not abort at step 4. Then, the bounds $B_{r,i}, B_{r,e}$ are set so as to ensure that the proof generation in step 12 always succeeds except if $\|\mathbf{r}_e\|_2 > B_{r,e}$, which happens with negligible probability. The verification at step 14 also goes through due to the completeness of the proof system. This holds true for the checks performed on the partial signature $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ thanks to the correctness of the signature scheme from [AGJ+24]. Finally, we have to show that the decomposition and bounds are correct for the second argument.

First, we have $\mathbf{w}_{1,L} + b_1\mathbf{w}_{1,H} = \mathsf{Low}(\mathbf{v}_1 - \mathbf{r}_{1,L}, b_1) + b_1\mathsf{High}(\mathbf{v}_1 - \mathbf{r}_{1,L}, b_1) - b_1\mathbf{r}_{1,H} = \mathbf{v}_1 - (\mathbf{r}_{1,L} + b_1\mathbf{r}_{1,H}) = \mathbf{v}_1 - \mathbf{r}_1$. Similarly, we have $\mathbf{w}_{i,L} + b_i\mathbf{w}_{i,H} = \mathbf{v}_i - \mathbf{r}_i$ for $i \in \{2,3\}$. We thus have

$$\begin{aligned} &b_1\mathbf{A}\mathbf{w}_{1,H} + \mathsf{t}\mathbf{G}\mathbf{w}_{2,L} + b_2\mathsf{t}\mathbf{G}\mathbf{w}_{2,H} - b_2\mathbf{B}\mathbf{w}_{2,H} + b_3\mathbf{A}_3\mathbf{w}_{3,H} \\ &\quad = \mathbf{u} + \mathbf{d}\cdot\mathcal{H}(m) - \mathbf{A}\mathbf{w}_{1,L} + \mathbf{B}\mathbf{w}_{2,L} - \mathbf{A}_3\mathbf{w}_{3,L} \bmod qR \end{aligned}$$

Then, it holds that $\|\mathbf{w}_{1,H}\|_2 \le (B_1 + \|\mathbf{r}_1\|_2 + \|\mathbf{w}_{1,L}\|_2)/b_1 \le B_1'$, because $\mathbf{w}_{1,L}$ has entries in $\widetilde{S}_{b_1}$, and $\mathbf{r}_1$ in $\widetilde{S}_{2b_1}$. Additionally, we have $\|[\mathbf{w}_{2,H}|\mathbf{w}_{3,H}]\|_2 \le (B_2 + \|[\mathbf{r}_2|\mathbf{r}_3]\|_2 + \|[\mathbf{w}_{2,L}|\mathbf{w}_{3,L}]\|_2)/b_2 \le B_2'$, as $\mathbf{r}_i, \mathbf{w}_{i,L}$ has entries in $\widetilde{S}_{b_2}$ for $i \in \{2,3\}$. Generation and verification of the second argument then always succeeds.

□

4.2 One-More Unforgeability

Theorem 4.1. *The blind signature of Algorithms 3.1, 3.2, 3.3 and 3.4 is one-more unforgeable based on the hardness of* $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$, $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$, $\text{M-SIS}_{n,d,2d+2,q,\beta_{❶}}$, $\text{M-SIS}_{n,d,2d,q,\beta_{❷}}$, *the collision resistance of* $\mathcal{H}$*, and the soundness of* $(\mathsf{Prove}_1, \mathsf{Verify}_1)$ *and* $(\mathsf{Prove}_2, \mathsf{Verify}_2)$*. More precisely, the advantage of* PPT *adversary in breaking the one-more unforgeability of the blind signature is upper-bounded by*

$$\begin{aligned} &Adv_{OM\text{-}UF}[\mathcal{A}] \\ &\lesssim \varepsilon^{(e)}_{\text{M-LWE}} + \varepsilon_{cr}(\mathcal{H}) + \varepsilon^{(2)}_{sound} + 2\max\left(h^{\circ d}\left(C(|\mathcal{T}_w| - Q)\varepsilon^{❶}_{\text{M-SIS}} + k\varepsilon^{(1)}_{\text{M-LWE}}\right),\right. \\ &\quad \left.\varepsilon^{(1)}_{\text{M-LWE}} + \frac{1+\varepsilon}{1-\varepsilon}\left(\varepsilon^{(1)}_{sound} + 4CM_1M_2h^{\circ d}\left(CQ\varepsilon^{❷}_{\text{M-SIS}} + k\varepsilon^{(1)}_{\text{M-LWE}}\right)\right)\right) \end{aligned}$$

where $\varepsilon_{sound}^{(i)}$ is the soundness error of $(\mathsf{Prove}_i, \mathsf{Verify}_i)$ from Lemma A.2 and A.3, $\varepsilon_{cr}(\mathcal{H})$ is the advantage of $\mathcal{A}$ against the collision resistance of the hash function $\mathcal{H}$, $\varepsilon_{\text{M-LWE}}^{(e)}$ and $\varepsilon_{\text{M-LWE}}^{(1)}$ are the respective hardness bounds of $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ *and* $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$*, and $\varepsilon_{\text{M-SIS}}^{❶}$ and $\varepsilon_{\text{M-SIS}}^{❷}$ are the respective hardness bounds of* $\text{M-SIS}_{n,d,2d+m+1,q,\beta_{❶}}$ *and* $\text{M-SIS}_{n,d,2d,q,\beta_{❷}}$*. The constant $C \approx 2$ is the one from Lemma 2.1. The function $h^{\circ d}$ corresponds to the d-th composition power of the function h defined by*

$$h(x) = k\varepsilon_{\text{M-LWE}}^{(1)} + \delta\left(2k\varepsilon_{\text{M-LWE}}^{(1)} + \delta\left(k\varepsilon_{\text{M-LWE}}^{(1)} + x\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}},$$

with

$$\delta = 1 + Q(\lambda - 1/2)\cdot\left(\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{12d(n-1)+5}\left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{2ndk} - 1\right)^2$$

Proof. At a high level, the proof changes the one-more unforgeability game as follows. The challenger guesses the tag $\mathbf{t}^\star$ that will be included in one of the $Q+1$ produced signatures, hoping that it will actually correspond to a fresh adversarial signature and not a legitimately issued one. Depending on this guess, the challenger will craft the cryptographic material following either the type ❶ or type ❷ approach from [AGJ+24], while also hiding a decryption key in $\mathbf{b}_e$. For each of the Q emitted signatures, the challenger stores the digests[2] h'_i decrypted from the ciphertexts. Upon reception of the $Q+1$ message-signature $(m_i, \mathsf{sig}_i)_{i\in[Q+1]}$ pairs, the challenger chooses a pair (m_i, sig_i) such that $\mathcal{H}(m_i)$ does not belong to the set of decrypted digests mentioned above. Note that such an index i necessarily exists under the collision resistance of the function $\mathcal{H}$. It then extracts the proof included in sig_i, which can be used to solve an M-SIS instance if the included tag is $\mathbf{t}^\star$. We now prove it formally.

We denote G_0 to be the original game. For each game G, we call $\text{Adv}_G[\mathcal{A}]$ the advantage of $\mathcal{A}$ in winning the one-more unforgeability game in the modified setting of game G.

Game G_1. We first change the encryption material so that the challenger can embed a secret decryption key. More precisely, the challenger samples $\mathbf{A}_e \hookleftarrow U(R_p^{m_e\times d_e})$, $(\mathbf{s}_e, \mathbf{e}_e) \hookleftarrow \mathcal{B}_{\eta_e}^{d_e+m_e}$ and set $\mathbf{b}_e = \mathbf{A}_e\mathbf{s}_e + \mathbf{e}_e \bmod pR$. Then, when $\mathcal{A}$ queries $\mathcal{H}$ on $(\mathsf{seed}, \mathsf{domain\_separator}_5)$ and $(\mathsf{seed}, \mathsf{domain\_separator}_6)$, it reprograms the random oracle and outputs $\mathbf{A}_e$ and $\mathbf{b}_e$ respectively. A distinguisher between G_0 and G_1 can then be turned into a distinguisher for $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ on the instance $(\mathbf{A}_e, \mathbf{b}_e)$. When receiving an instance $(\mathbf{A}_e, \mathbf{b}_e)$, the M-LWE distinguisher would program $\mathcal{H}$ to output $\mathbf{A}_e$ and $\mathbf{b}_e$ as above. As such, if $\mathbf{b}_e$ is uniform, it perfectly simulates G_0 in the random oracle model, and if

[2] We insist that because the encryption and proof generations are adversarial, the adversary does not need to know an underlying input for these digests. As such we use h'_i and h^+ later for these decrypted/extracted elements instead of $\mathcal{H}(m'_i)$ and $\mathcal{H}(m^+)$ to avoid confusion.

$\mathbf{b}_e = \mathbf{A}_e\mathbf{s}_e + \mathbf{e}_e \bmod pR$, it simulates G_1. The distinguisher between G_0 and G_1 can then be used to distinguish between these two cases. As a result, we get

$$\mathrm{Adv}_{G_0}[\mathcal{A}] \leq \mathrm{Adv}_{G_1}[\mathcal{A}] + \varepsilon^{(e)}_{\text{M-LWE}},$$

where $\varepsilon^{(e)}_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$.

Game G_2. We now generate all the tags $\{\mathsf{t}^{(i)}; i \in [Q]\}$ at the outset of the game instead of at each signature issuance. The challenger also uses $\mathbf{s}_e$ to decrypt the ciphertext in each of the (at most) Q queries. More precisely, it computes $u^{(i)} = \mathsf{ct_1}^{(i)} - \mathbf{s}_e^T\mathsf{ct_0}^{(i)} \bmod pR$ and defines the coefficients of $h'_i \in T_1$ to be 0 if the corresponding coefficient of $u^{(i)}$ is closer to 0 than to $\lfloor p/2 \rceil$, and 1 otherwise. The challenger then stores the pairs $(h'_i, \mathsf{t}^{(i)})$ in a table T. We note that at this stage, the decrypted digests are not used. Additionally, as the ciphertexts are adversarially chosen, the decrypted digests may not be those which have been used in the commitment and zero-knowledge proofs. Because these changes do not change the view of the adversary, it then holds that G_2 is identically distributed as G_1.

Game G_3. In this game, the challenger proceeds as in G_2 and thus eventually receives $((m_i, \mathbf{w}_L^{(i)}, \pi_2^{(i)}))_{i\in[Q+1]}$ where all m_i are distinct. It then computes the corresponding digests $h_i = \mathcal{H}(m_i)$ and aborts if a collision occurs. In the latter case, $\mathcal{A}$ can readily be used against the collision resistance of $\mathcal{H}$ and we thus get

$$\mathrm{Adv}_{G_2}[\mathcal{A}] \leq \mathrm{Adv}_{G_3}[\mathcal{A}] + \varepsilon_{\mathrm{cr}}(\mathcal{H}).$$

Game G_4. We proceed as in G_3. At the end of the game, the challenger receives $((m_i, \mathbf{w}_L^{(i)}, \pi_2^{(i)}))_{i\in[Q+1]}$ from the adversary where all the proofs $\pi_2^{(i)}$ verify and all the m_i are distinct. Since G_3, there is no collision among the $\mathcal{H}(m_i)$ either. The challenger then computes $j = \min\{i \in [Q+1] : \mathcal{H}(m_i) \notin \mathsf{T}\}$. Then, except with the soundness error $\varepsilon^{(2)}_{\text{sound}}$ of $(\mathsf{Prove}_2, \mathsf{Verify}_2)$ stated in Lemma A.3, it extracts $\pi_2^{(j)}$ and gets $(\mathbf{w}^\star_{1,H}, \mathbf{w}^\star_{2,H}, \mathbf{w}^\star_{3,H}, \mathsf{t}^\star)$ such that

$$\begin{aligned} b_1\mathbf{A}\mathbf{w}^\star_{1,H} + \mathsf{t}^\star\mathbf{G}\mathbf{w}^{(j)}_{2,L} + b_2\mathsf{t}^\star\mathbf{G}\mathbf{w}^\star_{2,H} - b_2\mathbf{B}\mathbf{w}^\star_{2,H} + b_3\mathbf{A}_3\mathbf{w}^\star_{3,H} \\ = \mathbf{u} + \mathbf{d}\cdot\mathcal{H}(m_j) - \mathbf{A}\mathbf{w}^{(j)}_{1,L} + \mathbf{B}\mathbf{w}^{(j)}_{2,L} - \mathbf{A}_3\mathbf{w}^{(j)}_{3,L} \bmod qR, \\ \left\|\mathbf{w}^\star_{1,H}\right\|_2^2 \leq {B'_1}^2, \quad \left\|[\mathbf{w}^\star_{2,H}|\mathbf{w}^\star_{3,H}]\right\|_2^2 \leq {B'_2}^2, \quad \text{and} \quad \mathsf{t}^\star \in \mathcal{T}_w. \end{aligned}$$

It then holds that

$$\mathrm{Adv}_{G_3}[\mathcal{A}] \leq \mathrm{Adv}_{G_4}[\mathcal{A}] + \varepsilon^{(2)}_{\text{sound}}.$$

Game G_5. We now introduce the branching of our security reduction. At the outset, the challenger samples $\rho \hookleftarrow U(\{1,2\})$. If $\rho = 1$, the challenger expects what we call a type ❶ forgery which corresponds to a one-more forgery where the extracted tag $\mathsf{t}^\star$ from G_4 is not in $\{\mathsf{t}^{(i)}; i \in [Q]\}$. On the other hand, if $\rho = 2$, the challenger expects a type ❷ forgery which corresponds to a one-more forgery where the extracted tag is among the emitted ones, i.e., there exists $i \in [Q]$ such that $\mathsf{t}^\star = \mathsf{t}^{(i)}$. The reduction aborts if the guess on the type of the forgery is wrong. As a result, we get

$$\mathrm{Adv}_{G_4}[\mathcal{A}] = 2\mathrm{Adv}_{G_5}[\mathcal{A}].$$

Based on the value of ρ the challenger will then set the cryptographic materials and queries differently. We define $\mathrm{Adv}_G^{❶}[\mathcal{A}]$ to be the advantage of $\mathcal{A}$ in returning a type ❶ one-more forgery in game G, and $\mathrm{Adv}_G^{❷}[\mathcal{A}]$ for a type ❷ one-more forgery. Because in G_5 the type guess is correct, it holds that $\mathrm{Adv}_{G_5}[\mathcal{A}] \leq \max(\mathrm{Adv}_G^{❶}[\mathcal{A}], \mathrm{Adv}_G^{❷}[\mathcal{A}])$. We later introduce changes that may be specific to one branch only. When describing each game, we specify which branch is impacted (❶ or ❷), except if the change impacts both branches.

The first change we introduce is that after sampling ρ and the set of tags, the challenger samples $\mathsf{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathsf{t}^{(i)}; i \in [Q]\})$ in branch ❶, and in branch ❷ it samples $i^+ \hookleftarrow U([Q])$ and sets $\mathsf{t}^+ = \mathsf{t}^{(i^+)}$. Because t^+ is not used anywhere yet, it does not change the advantage. Hence, it still holds that

$$\mathrm{Adv}_{G_4}[\mathcal{A}] \leq 2\max\left(\mathrm{Adv}_{G_5}^{❶}[\mathcal{A}], \mathrm{Adv}_{G_5}^{❷}[\mathcal{A}]\right)$$

<u>Game G_6 (❷).</u> In branch ❷, the challenger hides a short relation in $\mathbf{d}$. More precisely, it samples $\mathbf{s}$ from $\mathcal{B}_1^{2d}$ and defines $\mathbf{d} = \mathbf{A}\mathbf{s} \bmod qR$. It then reprograms $\mathcal{H}$ so as to output $\mathbf{d}$ when queried on $(\mathsf{seed}, \mathsf{domain\_separator}_1)$. Similarly to the change in G_1, we argue this change based on the $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$ assumption. As it only impacts the branch $i^+ \leq Q$, we have

$$\mathrm{Adv}_{G_5}^{❶}[\mathcal{A}] = \mathrm{Adv}_{G_6}^{❶}[\mathcal{A}], \text{ and } \mathrm{Adv}_{G_5}^{❷}[\mathcal{A}] \leq \mathrm{Adv}_{G_6}^{❷}[\mathcal{A}] + \varepsilon_{\text{M-LWE}}^{(1)},$$

where $\varepsilon_{\text{M-LWE}}^{(1)}$ is the hardness bound of $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$.

<u>Game G_7 (❷).</u> We hide a short relation in $\mathbf{u}$ in branch ❷. The challenger samples $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{2d},s_1}$ and $[\mathbf{v}_2^T|\mathbf{v}_3^T]^T \hookleftarrow \mathcal{D}_{R^{k(d+1)},s_2}$, and computes $\mathbf{u} = \mathbf{A}\mathbf{v}_1 + (\mathsf{t}^+\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 \bmod qR$. It then reprograms $\mathcal{H}$ so that it outputs $\mathbf{u}$ when queried on $(\mathsf{seed}, \mathsf{domain\_separator}_4)$. We use the same argument as from [AGJ+24] by relying on the regularity lemma of [GPV08, Cor. 5.2] recalled in Lemma 2.2. With our parameter choices, we then get that

$$\mathrm{Adv}_{G_6}^{❶}[\mathcal{A}] = \mathrm{Adv}_{G_7}^{❶}[\mathcal{A}], \text{ and } \mathrm{Adv}_{G_6}^{❷}[\mathcal{A}] \in \left[\frac{1}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right]\mathrm{Adv}_{G_7}^{❷}[\mathcal{A}],$$

<u>Game G_8 (❷).</u> In branch ❷, the tag t^+ is chosen among the tags used in the signing queries. As our goal will be to hide t^+ in the public key while still being able to answer signing queries, we need to change the query i^+. All other queries can be answered legitimately. Our goal is to use the hidden relation of $\mathbf{u}$ to answer the i^+-th query later. But it also needs to be a correct partial signature, which means we need to extract the message digest and commitment randomness

from $\pi_1^{(i^+)}$. In this game, we thus use the extractor from the soundness property of $(\mathsf{Prove}_1, \mathsf{Verify}_1)$. It thus obtains $(\mathbf{r}_1^+, \mathbf{r}_2^+, \mathbf{r}_3^+, h^+, \mathbf{r}_e^+)$ such that

$$\mathbf{A}\mathbf{r}_1^+ + (\mathsf{t}^+\mathbf{G} - \mathbf{B})\mathbf{r}_2^+ + \mathbf{A}_3\mathbf{r}_3^+ + \mathbf{d}h^+ = \mathbf{c}^{(i^+)} \bmod qR,$$
$$\mathbf{A}_e^T\mathbf{r}_e^+ = \mathsf{ct_0}^{(i^+)} \bmod pR, \quad \mathbf{b}_e^T\mathbf{r}_e^+ + \lfloor p/2\rceil h^+ = \mathsf{ct_1}^{(i^+)} \bmod pR,$$
$$\left\|\mathbf{r}_1^+\right\|_2 \leq B_{r,1}, \quad \left\|[\mathbf{r}_2^+|\mathbf{r}_3^+]\right\|_2 \leq B_{r,2}, \quad \left\|\mathbf{r}_e^+\right\|_2 \leq B_{r,e}, \quad h^+ \in T_1$$

except with a soundness error $\varepsilon_{\text{sound}}^{(1)}$, detailed in Lemma A.2.

Correct decryption. Notice here that the verifiable encryption proof ensures that the decrypted digest h'_{i^+} matches the one (h^+) that is in the commitment and thus actually signed by the challenger. Indeed, when decrypting, it holds that h'_{i^+} is obtained by decoding $\mathsf{ct_1}^{(i^+)} - \mathbf{s}_e^T\mathsf{ct_0}^{(i^+)} = \mathbf{e}_e^T\mathbf{r}_e^+ + \lfloor p/2\rceil h^+ \bmod pR$. Yet, it holds that $\left\|\mathbf{e}_e^T\mathbf{r}_e^+\right\|_\infty \leq \|\mathbf{e}_e\|_2\|\mathbf{r}_e\|_2 \leq \|\mathbf{e}_e\|_2 \cdot B_{r,e}$. Then, as $\mathbf{e}_e$ is drawn from $\mathcal{B}_{\eta_e}^{m_e}$, it holds that its norm is bounded with overwhelming probability by $B_{r,e}$ as well. Because we choose $p > 4B_{r,e}^2$, we get that the decryption is correct if the tail bound on $\mathbf{e}_e$ is verified. We insist here that because $\mathbf{r}_e$ is adversarial, we can only deduce a norm bound from the proof extraction but not a particular distribution. In particular, one cannot set p based on a bound on the inner product of two binomial vectors (which would be smaller) and needs to use this worst-case bound on $\mathbf{r}_e$. This correct decryption will be needed to exploit the forgery in the final game.

We then have

$$\mathrm{Adv}_{G_7}^{❶}[\mathcal{A}] = \mathrm{Adv}_{G_8}^{❶}[\mathcal{A}], \text{ and } \mathrm{Adv}_{G_7}^{❷}[\mathcal{A}] \leq \mathrm{Adv}_{G_8}^{❷}[\mathcal{A}] + \varepsilon_{\text{sound}}^{(1)},$$

Game G_9 (❷). We now enforce a bound on $\mathbf{s} \cdot h^+$ in anticipation of the rejection sampling argument in the next games. We use the heuristic bound from [AGJ+24, Lem. 2.4] recalled in Lemma 2.1. Concretely, for the i^+-th query in branch ❷, after extracting h^+ we abort the reduction altogether if $\|\mathbf{s}h^+\|_2 > \sqrt{nd}\|h^+\|_2$. It then holds that

$$\mathrm{Adv}_{G_8}^{❶}[\mathcal{A}] = \mathrm{Adv}_{G_9}^{❶}[\mathcal{A}], \text{ and } \mathrm{Adv}_{G_8}^{❷}[\mathcal{A}] \leq C\mathrm{Adv}_{G_9}^{❷}[\mathcal{A}],$$

for a small constant $C \approx 2$.

Game G_{10} (❷). We further prepare the rejection sampling argument for the i^+-th query only. During the i^+-th query, the challenger samples $(\mathbf{v}_1^{(+)}, \mathbf{v}_2^{(+)}, \mathbf{v}_3^{(+)})$ legitimately using the preimage sampler, and then rejects based on the value of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ which are so far independent of $(\mathbf{v}_1^{(+)}, \mathbf{v}_2^{(+)}, \mathbf{v}_3^{(+)})$. More precisely, it samples $u_1, u_2 \hookleftarrow U([0,1))$. The reduction continues only if $u_1 \leq 1/M_1$, $u_2 \leq 1/M_2$, and $\langle \mathbf{v}_1\ ,\ \mathbf{s}h^+ + \mathbf{r}_1^+\rangle \geq 0$ and $\langle [\mathbf{v}_2|\mathbf{v}_3]\ ,\ [\mathbf{r}_2^+|\mathbf{r}_3^+]\rangle \geq 0$, otherwise the challenger aborts. We insist that if the challenger does not abort, it answers the query with $(\mathbf{v}_1^{(+)}, \mathbf{v}_2^{(+)}, \mathbf{v}_3^{(+)})$ and not the hidden relation of $\mathbf{u}$. It then holds that

$$\mathrm{Adv}_{G_9}^{❶}[\mathcal{A}] = \mathrm{Adv}_{G_{10}}^{❶}[\mathcal{A}], \text{ and } \mathrm{Adv}_{G_9}^{❷}[\mathcal{A}] \leq 4M_1M_2\mathrm{Adv}_{G_{10}}^{❷}[\mathcal{A}],$$

Game G_{11} (❷). We now use the relation hidden in $\mathbf{u}$ to answer the i^+-th query in branch ❷, and perform rejection sampling to ensure a correct distribution so that the adversary does not notice. After extracting $\pi_1^{(i^+)}$, the challenger samples $u_1, u_2 \hookleftarrow U([0,1))$, computes $\delta_1 = \langle \mathbf{v}_1 + \mathbf{s}h^+ + \mathbf{r}_1^+ \ , \ \mathbf{s}h^+ + \mathbf{r}_1^+ \rangle$ and $\delta_2 = \langle [\mathbf{v}_2 + \mathbf{r}_2^+ | \mathbf{v}_3 + \mathbf{r}_3^+] \ , \ [\mathbf{r}_2^+ | \mathbf{r}_3^+] \rangle$. It then aborts the reduction if

$$\delta_1 < 0, \text{ or } u_1 > \frac{1}{M_1} \exp\left(\frac{\pi}{s_1^2} \left(\left\| \mathbf{s}h^+ + \mathbf{r}_1^+ \right\|_2^2 - 2\delta_1 \right) \right),$$
$$\text{or } \delta_2 < 0, \text{ or } u_2 > \frac{1}{M_2} \exp\left(\frac{\pi}{s_2^2} \left(\left\| [\mathbf{r}_2^+ | \mathbf{r}_3^+] \right\|_2^2 - 2\delta_2 \right) \right).$$

If the challenger did not abort, it constructs

$$\begin{bmatrix} \mathbf{v}_{1,1}^{(i^+)} \\ \mathbf{v}_{1,2}^{(i^+)} \end{bmatrix} = \mathbf{v}_1 + \mathbf{s}h^+ + \mathbf{r}_1^+, \text{ and } \mathbf{v}_2^{(i^+)} = \mathbf{v}_2 + \mathbf{r}_2^+, \mathbf{v}_3^{(i^+)} = \mathbf{v}_3 + \mathbf{r}_3^+,$$

and outputs $(\mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}, \mathbf{v}_3^{(i^+)})$ as the partial signature in step 18 of Algorithm 3.3. Based on the bounds proven on the $\mathbf{r}_i^+$, the one enforced on $\mathbf{s}h^+$, and the way the parameters s_1, s_2 are set in Algorithm 3.1, the rejection sampling argument of Lemma 2.5 shows that the view of the adversary is identically distributed in G_{11} and G_{10}.

Hybrid Games $G_{j,i}$. For both branches, we now use the hybrid argument used in [AGJ+24] to hide the tag $\mathfrak{t}^+$ in the public key $\mathbf{B}$. The authors rely on a specific trapdoor switching method and define hybrid games $G_{j,i}$ for $j \in [d]$ and $i \in [0,9]$. More precisely, $G_{j,0}$ is essentially G_{11} but where $\mathbf{B} = [\mathbf{A}\mathbf{R}_1 + \mathfrak{t}^+\mathbf{G}_1 | \dots | \mathbf{A}\mathbf{R}_{j-1} + \mathfrak{t}^+\mathbf{G}_{j-1} | \mathbf{A}\mathbf{R}_j | \dots | \mathbf{A}\mathbf{R}_d]$ with $\mathbf{G}_\ell = \mathbf{e}_\ell \otimes [1|b|\dots|b^{k-1}] \in R^{d \times k}$ where $\mathbf{e}_\ell$ is the zero vector that has a 1 at the ℓ-th entry. In particular, we note that $G_{1,0} = G_8$. Then, $G_{j,1}$ hides a gadget in $\mathbf{A}_3$ as $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3'$ for $\mathbf{A}_3'$ drawn uniformly, thus reprogramming $\mathcal{H}$ on input $(\mathsf{seed}, \mathsf{domain\_separator}_3)$. In $G_{j,2}$, the challenger hides a relation in $\mathbf{A}_3$ as $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}\mathbf{R}_j'$ under M-LWE$^k_{n,d,d,q,\mathcal{B}_1}$. In $G_{j,3}$, the challenger uses the partial trapdoor $\mathbf{R}_j'$ instead of $\mathbf{R}_j$ to produce signatures (except for the i^+-th query, if $i^+ \leq Q$, which remains unchanged), which is argued by the trapdoor switching lemma of [AGJ+24, Lem. 4.1] recalled in Lemma 2.4. Then, $G_{j,4}$ simulates the partial public key $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j \bmod qR$ and instead samples $\mathbf{B}_j$ uniformly in $R_q^{d \times k}$, which is unbeknownst to $\mathcal{A}$ under M-LWE$^k_{n,d,d,q,\mathcal{B}_1}$. $G_{j,5}$ adds the tag guess as $\mathbf{B}_j = \mathbf{B}_j' + \mathfrak{t}^+\mathbf{G}_j$ with $\mathbf{B}_j'$ uniform. In $G_{j,6}$, it re-introduces a partial secret key to get $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j + \mathfrak{t}^+\mathbf{G}_j$ under M-LWE$^k_{n,d,d,q,\mathcal{B}_1}$. The trapdoor switching is used again in $G_{j,7}$ to use the partial trapdoor $\mathbf{R}_j$ instead of $\mathbf{R}_j'$. In $G_{j,8}$, we replace $\mathbf{A}\mathbf{R}_j'$ by a uniform $\mathbf{A}_3'$ again under M-LWE$^k_{n,d,d,q,\mathcal{B}_1}$. Finally, $\mathbf{A}_3$ is again changed to be perfectly uniform in $G_{j,9}$ so that $G_{j,9} = G_{j+1,0}$.

The analysis of the hybrid argument is exactly the same as in [AGJ+24], and it thus holds that by looping over $j \in [d]$, we have

$$\mathrm{Adv}^{❶}_{G_{11}}[\mathcal{A}] \lesssim h^{\circ d}\left(\mathrm{Adv}^{❶}_{G_{d,9}}[\mathcal{A}]\right), \text{ and } \mathrm{Adv}^{❷}_{G_{11}}[\mathcal{A}] \lesssim h^{\circ d}\left(\mathrm{Adv}^{❷}_{G_{d,9}}[\mathcal{A}]\right),$$

where $h^{\circ d}$ corresponds to the d-th composition power of the function h, which is itself defined by

$$h(x) = k\varepsilon^{(1)}_{\text{M-LWE}} + \delta\left(2k\varepsilon^{(1)}_{\text{M-LWE}} + \delta\left(k\varepsilon^{(1)}_{\text{M-LWE}} + x\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}},$$

with

$$\delta = 1 + Q(\lambda - 1/2) \cdot \left(\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{12d(n-1)+5}\left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{2ndk} - 1\right)^2$$

Game G_{12}. For both branches, we re-introduce a short relation (without gadget) in $\mathbf{A}_3$. More precisely, it samples $\mathbf{R}'$ from $\mathcal{B}_1^{2d\times k}$ and defines $\mathbf{A}_3 = \mathbf{A}\mathbf{R}' \bmod qR$. It then reprograms the random oracle $\mathcal{H}$ so as to output $\mathbf{A}_3$ when queried on $(\mathsf{seed}, \mathsf{domain\_separator}_3)$. We can again argue this change based on the M-LWE$_{n,d,d,q,\mathcal{B}_1}$ assumption. We thus get

$$\mathrm{Adv}^{❶}_{G_{d,9}}[\mathcal{A}] \leq \mathrm{Adv}^{❶}_{G_{12}}[\mathcal{A}] + k\varepsilon^{(1)}_{\text{M-LWE}}, \text{ and } \mathrm{Adv}^{❷}_{G_{d,9}}[\mathcal{A}] \leq \mathrm{Adv}^{❷}_{G_{12}}[\mathcal{A}] + k\varepsilon^{(1)}_{\text{M-LWE}},$$

Game G_{13}. In this game, the challenger aborts if the tag guess is incorrect. More precisely, in branch ❶, the adversary must return a type ❶ one-more forgery which means that $\mathsf{t}^+ \notin \{\mathsf{t}^{(i)}; i \in [Q]\}$. Because t^+ is hidden to the view of the adversary, the guess is correct with probability $1/(|\mathcal{T}_w| - Q)$. For branch ❷, the adversary must return a type ❷ one-more forgery meaning there exists $i^\star \in [Q]$ such that $\mathsf{t}^\star = \mathsf{t}^{(i^\star)}$. The challenger thus aborts if $i^+ \neq i^\star$, meaning the guess is correct with probability $1/Q$. We thus get

$$\mathrm{Adv}^{❶}_{G_{12}}[\mathcal{A}] = (|\mathcal{T}_w| - Q)\mathrm{Adv}^{❶}_{G_{13}}[\mathcal{A}], \text{ and } \mathrm{Adv}^{❷}_{G_{12}}[\mathcal{A}] = Q\mathrm{Adv}^{❷}_{G_{13}}[\mathcal{A}],$$

Exploiting the one-more forgery. We now explain for each branch how to exploit the one-more forgery outputted by $\mathcal{A}$ to find a solution of a specific M-SIS instance. As the two branches are fairly different, we thus bound $\mathrm{Adv}^{❶}_{G_{13}}[\mathcal{A}]$ and $\mathrm{Adv}^{❷}_{G_{13}}[\mathcal{A}]$ separately. The challenger indeed receives two instances $\overline{\mathbf{A}} = [\mathbf{I}_d|\mathbf{A}'|\mathbf{d}|\mathbf{u}]$ and $\overline{\mathbf{A}}' = [\mathbf{I}_d|\mathbf{A}']$ of M-SIS$_{n,d,2d+2,q,\beta_{❶}}$ and M-SIS$_{n,d,2d,q,\beta_{❷}}$ respectively. The first will be used to define the material when $\rho = 1$, while the other will serve for the branch where $\rho = 2$. Depending on the value ρ sampled at the outset, it discards one of these two instances and reprograms $\mathcal{H}$ for the relevant material[3] from the M-SIS instance. It then proceeds as in G_{13} following the determined branch.

Branch ❶. We start with the easier branch to bound $\mathrm{Adv}^{❶}_{G_{13}}[\mathcal{A}]$. It holds that $\mathsf{t}^\star = \mathsf{t}^+$, which means that $\mathsf{t}^\star\mathbf{G} - \mathbf{B} = \mathsf{t}^\star\mathbf{G} - \mathbf{A}\mathbf{R} - \mathsf{t}^+\mathbf{G} = -\mathbf{A}\mathbf{R} \bmod qR$. We define $\mathbf{w}_1 = \mathbf{w}^{(j)}_{1,L} + b_1\mathbf{w}^\star_{1,H}$ and $\mathbf{w}_{23} = [\mathbf{w}^{(j)}_{2,L}|\mathbf{w}^{(j)}_{3,L}] + b_2[\mathbf{w}^\star_{2,H}|\mathbf{w}^\star_{3,H}]$. The

[3] Recall that in the branch ❶, all the partial signature queries are answered legitimately using the preimage sampler, and there is no hidden relation in $\mathbf{d}$ nor $\mathbf{u}$.

challenger then aborts if $\|[-\mathbf{R}|\mathbf{R}']\mathbf{w}_{23}\|_2 > \sqrt{nd}\|\mathbf{w}_{23}\|_2$. Again, by the Johnson-Lindenstrauss-like bound of [AGJ+24, Lem. 2.4] recalled in Lemma 2.1 (as $[-\mathbf{R}|\mathbf{R}']$ is indeed drawn from $\mathcal{B}_1$), the challenger continues with probability at least $1/C$ for $C \approx 2$. We then re-write the extraction equation as

$$\mathbf{A}\mathbf{w}_1 + \mathbf{A}[-\mathbf{R}|\mathbf{R}']\mathbf{w}_{23} - \mathbf{d}\mathcal{H}(m_j) - \mathbf{u} = \mathbf{0} \bmod qR,$$

i.e., $[\mathbf{I}_d|\mathbf{A}'|\mathbf{d}|\mathbf{u}]\mathbf{x}^\star = \mathbf{0} \bmod qR$ with $\mathbf{x}^\star = [(\mathbf{w}_1 + [-\mathbf{R}|\mathbf{R}']\mathbf{w}_{23})^T| - \mathcal{H}(m_j)| - 1]^T$. The last coefficient is non zero which ensures $\mathbf{x}^\star \neq \mathbf{0}$. Based on the extracted value, it holds that $\|\mathbf{w}_1\|_2 \leq b_1\sqrt{2nd} + b_1 B_1'$, and $\|\mathbf{w}_{23}\|_2 \leq b_2\sqrt{nk(d+1)} + b_2 B_2'$. We thus have

$$\|\mathbf{x}^\star\|_2 \leq \sqrt{\left(b_1\left(\sqrt{2nd} + B_1'\right) + \sqrt{nd}\cdot b_2\left(\sqrt{nk(d+1)} + B_2'\right)\right)^2 + n + 1} = \beta_{❶},$$

thus proving that $\mathbf{x}^\star$ is a solution of M-SIS$_{n,d,2d+2,q,\beta_{❶}}$. We get $\mathrm{Adv}_{\text{M-SIS}}[\mathcal{A}] \geq \mathrm{Adv}^{❶}_{G_{13}}[\mathcal{A}]/C$ which leads to

$$\mathrm{Adv}^{❶}_{G_{13}}[\mathcal{A}] \leq C\varepsilon^{❶}_{\text{M-SIS}}.$$

Branch ❷. We now bound $\mathrm{Adv}^{❷}_{G_{13}}[\mathcal{A}]$. Recall we are also in the case where $\mathbf{t}^\star = \mathbf{t}^+$ so that $\mathbf{t}^\star\mathbf{G} - \mathbf{B} = -\mathbf{A}\mathbf{R} \bmod qR$. We define $\mathbf{w}_1$ and $\mathbf{w}_{23}$ as in the previous branch and thus obtain the same bounds on their norm from the extraction. We then define $\boldsymbol{\Delta}\mathbf{w}_1 = \mathbf{w}_1 - (\mathbf{v}_1^{(i^+)} - \mathbf{r}_1^+)$ where $\mathbf{v}_1^{(i^+)}$ was part of the partial signature in the i^+-th query, and $\mathbf{r}_1^+$ was the randomness extracted in the i^+-th query. We also define

$$\boldsymbol{\Delta}\mathbf{w}_{23m} = \begin{bmatrix} \mathbf{w}_{23} - \begin{bmatrix} \mathbf{v}_2^{(i^+)} - \mathbf{r}_2^+ \\ \mathbf{v}_3^{(i^+)} - \mathbf{r}_3^+ \end{bmatrix} \\ h^+ - \mathcal{H}(m_j) \end{bmatrix}.$$

The challenger then aborts if $\|[-\mathbf{R}|\mathbf{R}'|\mathbf{s}]\boldsymbol{\Delta}\mathbf{w}_{23m}\|_2 > \sqrt{nd}\|\boldsymbol{\Delta}\mathbf{w}_{23m}\|_2$ which happens with probability at most $1 - 1/C$ for $C \approx 2$ using Lemma 2.1 again. Then, it holds that $\mathbf{A}\mathbf{w}_1 + \mathbf{A}[-\mathbf{R}|\mathbf{R}']\mathbf{w}_{23} - \mathbf{A}\mathbf{s}\mathcal{H}(m_j) = \mathbf{u} \bmod qR$. Yet, $\mathbf{u} = \mathbf{A}\mathbf{v}_1 - \mathbf{A}\mathbf{R}\mathbf{v}_2 + \mathbf{A}\mathbf{R}'\mathbf{v}_3 = \mathbf{A}(\mathbf{v}_1^{(i^+)} - \mathbf{s}h^+ - \mathbf{r}_1^+) - \mathbf{A}\mathbf{R}(\mathbf{v}_2^{(i^+)} - \mathbf{r}_2^+) + \mathbf{A}\mathbf{R}'(\mathbf{v}_3^{(i^+)} - \mathbf{r}_3^+) \bmod qR$. We can thus rewrite the equation as

$$\mathbf{A}\left(\boldsymbol{\Delta}\mathbf{w}_1 + [-\mathbf{R}|\mathbf{R}'|\mathbf{s}]\boldsymbol{\Delta}\mathbf{w}_{23m}\right) = \mathbf{0} \bmod qR.$$

Because $\mathcal{H}(m_j) \notin \mathcal{T}$, it holds that $\mathcal{H}(m_j) \neq h'_{i^+}$. Also, based on the decryption correctness explained in Game G_8, it holds that $h'_{i^+} = h^+$, and as such we have $\mathcal{H}(m_j) \neq h^+$. Using the same argument as in previous works, e.g., [LLM+16,LNPS21,LNP22,JRS23,AGJ+24], the unpredictability of $\mathbf{s}$ ensures that $\mathbf{x}^\star = \boldsymbol{\Delta}\mathbf{w}_1 + [-\mathbf{R}|\mathbf{R}'|\mathbf{s}]\boldsymbol{\Delta}\mathbf{w}_{23m}$ is non-zero except with negligible probability. We now bound $\mathbf{x}^\star$. Because of the rejection sampling argument in game G_{11}, it holds that $\mathbf{v}_1^{(i^+)} \sim \mathcal{D}_{R^{2d},s_1}$, and $[\mathbf{v}_2^{(i^+)}|\mathbf{v}_3^{(i^+)}] \sim \mathcal{D}_{R^{k(d+1)},s_2}$. We can then derive bounds on $\boldsymbol{\Delta}\mathbf{w}_1$ and $\boldsymbol{\Delta}\mathbf{w}_{23m}$ using the Gaussian tail bound of Lemma 2.3, and the bounds proven on the $\mathbf{r}_i^+$, as well as the facts that h^+ and $\mathcal{H}(m_j)$ are in T_1. In particular, we get $\|\boldsymbol{\Delta}\mathbf{w}_1\|_2 \leq b_1(\sqrt{2nd} + B_1') + B_1 + 2b_1\sqrt{2nd}$ $(\approx 2B_1 + 6b_1\sqrt{2nd})$, and similarly

$$\|\boldsymbol{\Delta}\mathbf{w}_{23m}\|_2 \leq \sqrt{(b_2(B_2' + \sqrt{nk(d+1)}) + B_2 + b_2\sqrt{nk(d+1)})^2 + n}.$$

In the end, we obtain

$$\begin{aligned}\|\mathbf{x}^\star\|_2 &\leq b_1\left(B_1' + 3\sqrt{2nd}\right) + B_1 + \sqrt{nd}\sqrt{\left(b_2\left(B_2' + 2\sqrt{nk(d+1)}\right) + B_2\right)^2 + n} \\ &= \beta_{❷},\end{aligned}$$

thus proving that $\mathbf{x}^\star$ is a solution of $\text{M-SIS}_{n,d,2d,q,\beta_{❷}}$. We obtain $\text{Adv}_{\text{M-SIS}}[\mathcal{A}] \geq \text{Adv}^{❷}_{G_{13}}[\mathcal{A}]/C - \mathsf{negl}(\lambda)$ which leads to

$$\text{Adv}^{❷}_{G_{13}}[\mathcal{A}] \leq C\varepsilon^{❷}_{\text{M-SIS}} + \mathsf{negl}(\lambda).$$

Advantage Bound. We can now combine all of the advantage bounds from each game hop from G_5, with the derived bounds on the advantages in G_{13}. We obtain the following equations.

$$\begin{aligned}\text{Adv}^{❶}_{G_5}[\mathcal{A}] &\lesssim h^{\circ d}\left(C(|\mathcal{T}_w| - Q)\varepsilon^{❶}_{\text{M-SIS}} + k\varepsilon^{(1)}_{\text{M-LWE}}\right), \\ \text{Adv}^{❷}_{G_5}[\mathcal{A}] &\lesssim \varepsilon^{(1)}_{\text{M-LWE}} + \frac{1+\varepsilon}{1-\varepsilon}\left(4CM_1M_2h^{\circ d}\left(CQ\varepsilon^{❷}_{\text{M-SIS}} + k\varepsilon^{(1)}_{\text{M-LWE}}\right) + \varepsilon^{(1)}_{\text{sound}}\right).\end{aligned}$$

Combining these inequalities with the first game hops from G_0 to G_5 gives the claimed bound on $\text{Adv}_{G_0}[\mathcal{A}]$.

□

4.3 Blindness

Theorem 4.2. *The blind signature of Algorithms 3.1, 3.2, 3.3 and 3.4 is blind in the random oracle model based on* $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$, $\text{M-LWE}_{n,d,d,q,U(\widetilde{T}_1)}$, *and the zero-knowledge property of* $(\mathsf{Prove}_1, \mathsf{Verify}_1)$ *and* $(\mathsf{Prove}_2, \mathsf{Verify}_2)$*. More precisely, the advantage of* PPT *adversary in breaking the blindness of the blind signature is upper-bounded by*

$$Adv_{blind}[\mathcal{A}] \leq 2(\varepsilon^{(1)}_{zk} + \varepsilon^{(2)}_{zk} + \varepsilon^{(e)}_{\text{M-LWE}} + \varepsilon^{(2)}_{\text{M-LWE}}),$$

where $\varepsilon^{(i)}_{zk}$ *is the zero-knowledge loss of* $(\mathsf{Prove}_i, \mathsf{Verify}_i)$ *from Lemma A.2 and A.3,* $\varepsilon^{(e)}_{\text{M-LWE}}$ *and* $\varepsilon^{(2)}_{\text{M-LWE}}$ *are the respective hardness bounds of* $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ *and* $\text{M-LWE}_{n,d,d,q,U(\widetilde{T}_1)}$.

Proof. We prove the blindness by a sequence of hybrids where we change the blindness game progressively while proving these changes can be made unbeknownst to the adversary. The game G_0 is the original blindness game from Definition 2.5. More precisely, the adversary chooses (potentially maliciously) the public parameters pp (which includes $\mathsf{seed} \in \{0,1\}^{256}$ to derive public matrices and vectors), the public key $\mathsf{pk} = \mathbf{B}$ and two messages m_0 and m_1 in $\{0,1\}^*$. It sends it to the challenger, who then samples a random bit $\rho \hookleftarrow U(\{0,1\})$. After that, the challenger plays the role of the user in two concurrent interactions of Sign with the adversary $\mathcal{A}$. In the end, it gets sig_0 as the blind signature on m_ρ and sig_1 as the blind signature on $m_{1-\rho}$ and sends both signatures to $\mathcal{A}$. The adversary then decide on a value $\rho^\star \in \{0,1\}$ and wins if $\rho = \rho^\star$.

Game G_1. In this game we first change the way hash queries are handled, especially those to derive the verifiable encryption materials. When $\mathcal{A}$ queries the random oracle on $(\mathsf{seed}, \mathsf{domain\_separator_i})$, the challenger samples the matrix or vector uniformly in its space. In particular, on input $(\mathsf{seed}, \mathsf{domain\_separator_2})$, $(\mathsf{seed}, \mathsf{domain\_separator_5})$ and $(\mathsf{seed}, \mathsf{domain\_separator_6})$, the challenger samples $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$, $\mathbf{A}_e \hookleftarrow U(R_p^{m_e\times d_e})$ and $\mathbf{b}_e \hookleftarrow U(R_p^{m_e})$ respectively. In the random oracle model, G_0 and G_1 are identical.

Game G_2. The challenger now simulates the zero-knowledge proofs involved in both executions of the blind signing protocols. Instead of calling Prove_1 and Prove_2 on the secret data, it calls the simulators $\mathcal{S}_1$ and $\mathcal{S}_2$ without resorting to the secret witnesses. Based on the zero-knowledge properties of both proof systems, Lemma A.2 and A.3 yield

$$|\mathrm{Adv}_{G_2}[\mathcal{A}] - \mathrm{Adv}_{G_1}[\mathcal{A}]| \leq 2(\varepsilon_{\mathrm{zk}}^{(1)} + \varepsilon_{\mathrm{zk}}^{(2)})$$

Game G_3. In this game, the challenger will further simulates the ciphertexts. For each of the two executions of the blind signature, instead of computing ct_0 and ct_1 legitimately, it samples $\mathbf{u}_0 \hookleftarrow U(R_p^{d_e})$ and $u_1 \hookleftarrow U(R_p)$ and define the ciphertexts as $\mathsf{ct}_0 = \mathbf{u}_0$ and $\mathsf{ct}_1 = u_1 + \lfloor p/2 \rceil \mathcal{H}(m_\rho) \bmod pR$ (or with $\mathcal{H}(m_{1-\rho})$ for the other execution). Noticing that the randomness $\mathbf{r}_e$ is no longer used in the proof generation, this change can be argued with the M-LWE$_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ assumption. This argument follows the one we would use to prove the IND-CPA security of the encryption scheme. Here, we already enforced the encryption key to be uniform (i.e., so that the adversary does not have a secret key to decrypt) which results in a slightly simpler argument.

More precisely, we can use a distinguisher $\mathcal{D}$ between G_2 and G_3 to distinguish a 2-secret M-LWE$_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ instance

$$\begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} [\mathbf{r}_e^{(\rho)} | \mathbf{r}_e^{(1-\rho)}]$$

from uniform. This instance follows the Knapsack formulation of M-LWE, which is equivalent to M-LWE$_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$ as shown for example in [BJRW23, Sec. 4.1][4].

[4] The loss incurred by the reduction is linked to the splitting of p. In our case, we can choose p to split into at most $n/4$ factors to keep this loss negligible.

Given a 2-secret instance $([\mathbf{u}_0^{(\rho)}|u_1^{(\rho)}], [\mathbf{u}_0^{(1-\rho)}|u_1^{(1-\rho)}])$, the distinguisher would proceed as in G_2 and G_3 except in the generation of the ciphertexts where $\mathsf{ct_0}^{(i)} = \mathbf{u}_0^{(i)}$ and $\mathsf{ct_1}^{(i)} = u_1^{(i)} + \lfloor p/2 \rceil \mathcal{H}(m_i) \bmod pR$. If the instance is uniform, it perfectly simulates G_3 and otherwise it simulates G_2. It then holds that

$$|\mathrm{Adv}_{G_3}[\mathcal{A}] - \mathrm{Adv}_{G_2}[\mathcal{A}]| \leq \sup_{\mathcal{D}} \mathrm{Adv}_{G_2,G_3}[\mathcal{D}] \leq 2\varepsilon_{\text{M-LWE}}^{(e)},$$

where $\varepsilon_{\text{M-LWE}}^{(e)}$ is the hardness bound of $\text{M-LWE}_{n,d_e,m_e,p,\mathcal{B}_{\eta_e}}$.

Game G_4. As the messages are now masked by a perfectly random and independent value, we can have the challenger sample the ciphertexts at random directly without using the messages. So it directly samples $\mathsf{ct_1}^{(i)} \hookleftarrow U(R_p)$. As $u_1^{(i)}$ was chosen independently of m_i, the distribution stays the same and G_3 and G_4 are then identical.

Game G_5. We now change the way the commitment is computed by essentially relying on its hiding property. By decomposing $\mathbf{c}$ into $b_1\mathbf{A}\mathbf{r}_{1,H} + (\mathbf{A}\mathbf{r}_{1,L} + (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{r}_2 + \mathbf{A}_3\mathbf{r}_3 + \mathbf{d}\mathcal{H}(m)) \bmod qR$, we replace $\mathbf{A}\mathbf{r}_{1,H}$ by a uniform vector. As $\mathbf{r}_{1,H}$ is independent from the rest, this can be done under the $\text{M-LWE}_{n,d,d,q,U(\widetilde{T}_1)}$ assumption. The reason for using only the high-order bits $\mathbf{r}_{1,H}$ and not the whole $\mathbf{r}_1$ is because $\mathbf{r}_{1,L}$ is also used in the masked part $\mathbf{w}_{1,L}$ revealed in the blind signatures. But $\mathbf{r}_{1,H}$ is used nowhere else as we simulate the second zero-knowledge proof. As a result, the challenger samples $\mathbf{t}^{(i)} \hookleftarrow U(R_q^d)$ for each execution and constructs $\mathbf{c}^{(i)} = b_1\mathbf{t}^{(i)} + (\mathsf{t}^{(i)}\mathbf{G} - \mathbf{B})\mathbf{r}_2^{(i)} + \mathbf{A}_3\mathbf{r}_3^{(i)} + \mathbf{d}\mathcal{H}(m_i)) \bmod qR$. Using the same reasoning as that of G_2-G_3 but for the $\text{M-LWE}_{n,d,2d,q,U(\widetilde{T}_1)}$ instance $[\mathbf{I}_d|\mathbf{A}']\mathbf{r}_{1,H} \bmod qR$, we get that

$$|\mathrm{Adv}_{G_5}[\mathcal{A}] - \mathrm{Adv}_{G_4}[\mathcal{A}]| \leq 2\varepsilon_{\text{M-LWE}}^{(2)},$$

where $\varepsilon_{\text{M-LWE}}^{(2)}$ is the hardness bound of $\text{M-LWE}_{n,d,d,q,U(\widetilde{T}_1)}$. Note that for that we need to ensure that $\mathbf{A}'$ is perfectly uniform which is the case due to our change in G_1. Also, notice that $\widetilde{T}_1$ does not include zero coefficients which may be unusual. Using the bijection $x \in \{0,1\} \mapsto 2x-1 \in \{-1,1\}$, a trivial reduction shows that $\text{M-LWE}_{n,d,d,q,U(\widetilde{T}_1)}$ is equivalent to $\text{M-LWE}_{n,d,d,q,U(T_1)}$, that is with binary secret/error.

Game G_6. We keep modifying the commitment by sampling $\mathbf{t}^{(i)} \hookleftarrow U(R_q^d)$ and computing $\mathbf{c}^{(i)} = \mathbf{t}^{(i)} + (\mathsf{t}^{(i)}\mathbf{G} - \mathbf{B})\mathbf{r}_2^{(i)} + \mathbf{A}_3\mathbf{r}_3^{(i)} + \mathbf{d}\mathcal{H}(m_i)) \bmod qR$. Because b_1 is a power-of-two less than q, it holds that $b_1 \in \mathbb{Z}_q^\times \subset R_q^\times$. As such, $b_1 U(R_q^d) = U(R_q^d)$ which proves that G_5 and G_6 are identically distributed.

Game G_7. As $\mathbf{t}^{(i)}$ is independent from the rest, we can simply sample $\mathbf{c}^{(i)}$ uniformly in R_q^d without using the messages. It then holds that G_6 and G_7 are identically distributed.

Game G_8. Now that the randomness $\mathbf{r}_{1,L}, \mathbf{r}_2, \mathbf{r}_3$ are used nowhere else, we complete our argument by sampling the $\mathbf{w}_{j,L}$ uniformly in $\widetilde{S}_{b_j}$ without resorting to the adversarially chosen signatures $\mathbf{v}_i$. So the challenger, for each of the two blind signature executions, samples $\mathbf{w}_{j,L}^{(i)}$ uniformly with entries in $\widetilde{S_{b_j}}$.

To argue that it is identically distributed, we rely on the fact that our decomposition preserves the invariance of the uniform distribution by shift. It means that $\mathsf{Low}(y+U([-b_j, b_j-1]), b_j) = U([-b_j, b_j-1])$ for any $y \in \mathbb{Z}$ by Lemma 2.7. Applying it coefficient-wise ensures that G_7 and G_8 are identically distributed.

The final blind signatures $(\mathbf{w}_{1,L}, \mathbf{w}_{2,L}, \mathbf{w}_{3,L}, \pi_2)$, as well as the issuance transcripts, no longer depend on the messages and elements chosen by the adversary. As a result, they are independent of the bit ρ which means the probability that $\mathcal{A}$ finds the correct bit is exactly 1/2, making its advantage 0. We then have $\mathrm{Adv}_{G_8}[\mathcal{A}] = 0$. By combining the above hybrids, it then holds that

$$\mathrm{Adv}_{\mathrm{blind}}[\mathcal{A}] \leq 2(\varepsilon_{\mathrm{zk}}^{(1)} + \varepsilon_{\mathrm{zk}}^{(2)} + \varepsilon_{\text{M-LWE}}^{(e)} + \varepsilon_{\text{M-LWE}}^{(2)})$$

as claimed. □

5 Performance

5.1 Parameter Selection

The soundness and zero-knowledge losses $\varepsilon_{\mathrm{sound}}^{(i)}$ and $\varepsilon_{\mathrm{zk}}^{(i)}$ rely on some M-SIS and M-LWE assumptions identified in Theorem 4.1 and 4.2. As a result, we have to estimate the hardness of each of them and compute the actual security by taking into account the reduction loss specified in these theorems. For the M-LWE assumptions, we use the lattice estimator [APS15] on the unstructured assumptions to determine the minimal BKZ block size B for lattice reduction attacks. We note that algebraic attacks [AG11,ACF+15] or combinatorial attacks [Wag02,BKW03] in our case are more expansive than those based on lattice reduction. We then estimate the hardness bound with $\varepsilon_{\text{M-LWE}} = 2^{-(0.292B+16.4)}$ classically and $\varepsilon_{\text{M-LWE}} = 2^{-(0.257B+16.4)}$ quantumly. For $\text{M-SIS}_{n,d,m,q,\beta}$ assumptions, standard attacks find a smaller-dimensional lattice of dimension $N \in [nd, nm]$ that minimizes the root Hermite factor $\delta_B = \beta^{1/N} q^{-nd/N^2}$, and then we use the formula by Chen [Che13] to find the BKZ block size from δ_B. More precisely, Chen [Che13] showed that under the Gaussian heuristic and the Geometric Series Assumption, the root Hermite factor δ_B of a BKZ-reduced basis with block size B is $\delta_B \approx (B(\pi B)^{1/B}/2\pi e)^{1/2(B-1)}$. Then again, we estimate the hardness bound of said M-SIS instance by $\varepsilon_{\text{M-SIS}} = 2^{-(0.292B+16.4)}$ classically and $\varepsilon_{\text{M-SIS}} = 2^{-(0.257B+16.4)}$ quantumly.

We thus search for parameters n, d, d_e, q, p and also the parameters of the proof systems to get the desired security when plugged into the loss functions and the plethora of assumptions considered in Theorem 4.1 and 4.2. The parameters detailed in Tables B.1, B.2, B.3, B.4 in Appendix B lead to the hardness bounds indicated in Table 5.1, which still ensures 126 bits of classical security for the one-more unforgeability and 125 for the blindness, despite the reduction loss.

For these parameters, we reach blind signatures of 40.47 KB, for a total issuance transcript of 58.86 KB. We give in Table 5.2 the sizes in KB of each

$\varepsilon^{❶}_{\text{M-SIS}}$	$\varepsilon^{❷}_{\text{M-SIS}}$	$\varepsilon^{(1)}_{\text{M-LWE}}$	$\varepsilon^{(2)}_{\text{M-LWE}}$	$\varepsilon^{(e)}_{\text{M-LWE}}$	$\varepsilon^{(1)}_{\text{sound}}$	$\varepsilon^{(2)}_{\text{sound}}$	$\varepsilon^{(1)}_{\text{zk}}$	$\varepsilon^{(2)}_{\text{zk}}$
$2^{-203.5}$	$2^{-182.2}$	$2^{-157.7}$	$2^{-147.2}$	$2^{-163.2}$	$2^{-128.2}$	$2^{-126.9}$	$2^{-127.3}$	2^{-127}

Table 5.1. Classical hardness bounds of the different assumptions for the signature, encryption and proof systems.

contribution. The public key of the signer is 53.94 KB but only needs to be transmitted once so we do not include it in the issuance transcript.

	Issuance Transcript					**Blind Signature**	
	$\|\mathsf{t}\|$	$\|\mathbf{c}\|$	$\|\mathsf{ct}\|$	$\|\pi_1\|$	$\|\mathbf{v}\|$	$\|\mathbf{w}_L\|$	$\|\pi_2\|$
	0.03 KB	3.59 KB	1.62 KB	44.91 KB	8.70 KB	5.38 KB	35.09 KB
Total	**58.85 KB**					**40.47 KB**	

Table 5.2. Size estimates for the issuance transcript between the user and signer, and the blind signature.

These estimates are obtained based on the parameters chosen as described above and thus depend on the reduction loss. In particular, in the case of one-more unforgeability, we need to set a bound Q on the maximal number of signatures per key pair. We set $Q = 2^{32}$ which means that the tag guess loss $|\mathcal{T}_w|$ is at least 2^{32}. In practice, the loss incurred via guessing arguments is sometimes implicitly not taken into account in the parameter selection. Actually, it is unclear whether previous papers on blind signatures account for such loss factors or not. In our case, we design our scheme so as to keep this loss acceptable and consider it when setting parameters. By discarding it, we could choose smaller parameters and end up with a transcript size of 52.41 KB and a blind signature of 35.72 KB.

5.2 Implementation Performance

One caveat of the blind signature in [BLNS23a] is that the issuance relies on general-purpose NIZKs which are very complex and very slow in practice, even though they lead to short proofs. Our system does not need to rely on such tools which then significantly improves the issuance phase. To demonstrate the concrete practicality of our blind signature, we propose a proof-of-concept implementation in C[5]. For that, we built upon the implementation of anonymous credentials of [AGJ+24] which is publicly available[6] and adapted it to our scheme and parameters. In particular, we updated the parameter-specific parts such as

[5] https://github.com/latticeblindsignature/lattice-blind-signature
[6] https://github.com/Chair-for-Security-Engineering/lattice-anonymous-credentials

expansion and sampling procedures. We also implemented the encryption mechanism modulo p, the decomposition functions, and obviously overhaul the whole implementation of zero-knowledge proofs as our relations are different. Regarding the latter, we point out that our scheme features the optimizations on the proof from [LNP22, Sec. 4.4, App. A] and [LN22] as opposed to [AGJ+24].

We benchmarked our implementation using a laptop featuring an Intel Core i7 12800H CPU running at 4.6 GHz. We used the same compilation options as [AGJ+24], namely `-O3 -march=native` using `gcc 11.4.0` with `pthread` disabled when building FLINT. The timings (in milliseconds) of each main step of the blind signature can be found in Table 5.3.

Round	Procedure	Reference	Time (ms)			
			mean	med	min	max
	key gen.	Alg. 3.2	673.873	498.793	451.248	1212.177
①	tag gen.	Alg. 3.3, Steps 1-2.	0.002	0.002	0.002	0.014
②	tag verify	Alg. 3.3, Step 4.	0.001	0.001	0.001	0.001
	commit	Alg. 3.3, Steps 5-8.	2.305	2.302	2.294	2.325
	encrypt	Alg. 3.3, Steps 9-11.	0.373	0.375	0.360	0.385
	embed_1	Alg. 3.3, Step 12.	0.674	0.672	0.629	0.848
	prove_1	Alg. 3.3, Step 12.	721.214	514.244	189.558	1774.477
③	verify_1	Alg. 3.3, Step 14.	121.494	121.556	118.736	125.494
	pre sign cmt.	Alg. 3.3, Steps 15-17.	73.859	73.617	73.264	75.445
	pre sig verify	Alg. 3.3, Steps 19-21.	2.119	2.116	2.105	2.153
	decomp.	Alg. 3.3, Steps 22-23.	0.081	0.080	0.077	0.095
	embed_2	Alg. 3.3, Step 24.	2.556	2.551	2.521	2.655
	prove_2	Alg. 3.3, Step 24.	669.633	386.768	141.606	1738.951
	verify_2		88.231	88.022	85.609	90.618
	verify	Alg. 3.4	90.775	90.569	89.105	93.165

Table 5.3. Benchmark results. Statistics over 100 executions. Where applicable, the key and message were randomized. High variance timings are due to rejection sampling.

The entire blind signature generation following Algorithm 3.3 can thus be performed in 1595 ms on average, mostly because of the generation of the two ZK proofs π_1 and π_2. Note that although Prove_2 is part of the blind signature generation, it no longer requires the signer. The total "online" time can thus be amortized to 925 ms on average. We insist that these timings are reported by our implementation which is not yet optimized. Vast improvements could be brought for example by the use of a parameter-specific arithmetic backend (so as to rely on stack allocations rather than heap allocations which are generally much slower), parallel hashing, vectorized computations leveraging the AVX2 instruction set, etc. Additionally, as we only prove quadratic lattice relations, improvements on the zero-knowledge proof framework itself would transfer to our scheme. We leave these optimizations as future work. Nevertheless, this provides a first milestone in the implementation of lattice-based blind signatures.

5.3 Comparison

As discussed in the introduction, the state-of-the art of post-quantum blind signatures is limited to a few constructions, in particular since the work by Hauck *et al.* [HKLN20] which invalidates all previous 3-round designs. While this paper introduced a secure 3-round construction, it was essentially a proof of concept that is clearly outperformed by our scheme. We thus choose to compare our scheme to the much more competitive two-round constructions in [AKSY22,dPK22,BLNS23a]. Obviously, this comparison has some limits but we believe it remains relevant as the spirit of our construction is closer to the one of two-round designs.

In our security assessment, we opted for a more realistic lattice sieving cost model but we note that certain previous works use the Core-SVP model instead which estimate the hardness bounds as $2^{-0.292B}$ instead of $2^{-(0.292B+16.4)}$. Nevertheless, they aim for a smaller security target which means that we achieve a similar security level to these works when using the Core-SVP model. The comparison is therefore given for a target classical security of 128 bits.

	Assumptions	Round	NIZK (iss.)	|transcript|	|bsig|
[AKSY22]	One-More-ISIS NTRU M-SIS/M-LWE	2	None	1.37 KB	45.19 KB
[dPK22]	NTRU M-SIS/M-LWE	2	Algebraic Relations	932 KB	102.6 KB
[BLNS23a]	NTRU M-SIS/M-LWE	2	General Purpose	60 KB	22 KB
Ours	M-SIS/M-LWE	3	Algebraic Relations	58.85 KB	40.47 KB

Table 5.4. Comparison of lattice-based blind signatures. The transcript size corresponds to all the messages exchanged between the signer and user. The "NIZK (iss.)" column describes the type of relations proven only in the issuance part, the proof being part of the transcript. The final generation of the blind signature includes a proof of algebraic lattice relations for all the constructions cited above.

Compared to [AKSY22], we achieve slightly smaller blind signatures at the expense of a larger transcript due to the commitment opening proof π_1. But most importantly, our construction shows that we can achieve the same level of compactness without compromising on security. Indeed, we do not need the one-more-ISIS assumption and only rely on the standard lattice assumptions M-SIS and M-LWE. Our blind signature and transcript sizes are also much smaller than the construction of [dPK22], whose security is based on standard assumptions as well. Finally, the blind signature of [BLNS23a] achieves a similar transcript size but with a blind signature that is almost twice as small as ours.

However, as mentioned in the introduction, their construction requires the use of general purpose NIZKs for the commitment opening proof which are very time-consuming. The author estimate this proof generation to be around 20 seconds. In our case, our issuance does not need to prove hash evaluations which means we can use efficient proof systems for algebraic lattice relations. Typically, the generation of π_1 takes on average 720 ms in our proof-of-concept implementation. As mentioned above, this timing can be vastly improved with a highly optimized implementation. All in all, our blind signature is competitive with the existing ones in terms of size, while providing strong security guarantees by relying solely on standard lattice assumptions, and by having a concretely efficient issuance.

6 Round-Optimal Blind Signature

The previous construction describes a blind signature in three rounds, the role of the first round being only to transfer the tag $\mathfrak{t}$ to use in the commitment phase. This 3-round version essentially allows us to rely on the stateful (adapted) version of the signature of [AGJ+24]. As such, it avoids certain intricacies of the security proof making the scheme more efficient. We insist however that as mentioned in Section 1.1, our 3-round blind signature of Section 3 is not subject to the ROS attack due to its very different structure compared to Schnorr-like blind signatures. Nevertheless, in this section, we describe how to simply obtain a round-optimal, i.e., 2-round, version of our scheme.

Actually, all we require from $\mathfrak{t}$ is that it differs for each signature issuance. For example, the tag $\mathfrak{t}$ could be randomly generated by the user and then sent, along with the commitment and the proof, to the signer. The latter would then check if $\mathfrak{t}$ has already been used in a previous session in which case it would abort. An alternative option, which does not require to maintain a state on the signer side, is the one where the user would deterministically derive the tag from some fresh public data using a hash function. For example, in the case of a user accessing to a remote signer, those data could be extracted from the previous messages exchanged to establish a secure channel (e.g. the TLS handshake).

In all cases, we need to limit the number of incidental tags collisions, which requires to work with a larger tag space and thus to adapt our protocol to limit the reduction loss. In the following, we consider the option where the tag is derived from a hash function since the other alternative can readily be derived from it.

Actually, our 3-round scheme follows a similar argument to the stateless version of [AGJ+24] described in the construction of [JRS23] from which it takes inspiration. We let ω be a positive integer smaller than n. In the first round, the user U starts by hashing a bitstring in (the public data mentioned above) of length l using an extendable output function such as SHAKE256) to obtain ω ring elements $\mathfrak{t}_1, \ldots, \mathfrak{t}_\omega$ in $\sum_{i \in [0, \lfloor n/\omega \rfloor - 1]} \{0, 1\} \cdot X^i$, that is elements of T_1 such that all the coefficients of degree higher than $\lfloor n/\omega \rfloor$ are 0. The signature matrix (also used for the commitment) is then

$$\mathbf{A}_{\mathsf{t}_1,\ldots,\mathsf{t}_\omega} = [\mathbf{A}| \sum_{j\in[\omega]} \mathsf{t}_j \mathbf{B}_j - \mathbf{B}|\mathbf{A}_3],$$

where the matrices $\mathbf{B}_i$ are appended to the public key and are defined as $\mathbf{B}_j = X^{(j-1)\lfloor n/\omega \rfloor}\mathbf{G} - \mathbf{A}\mathbf{R}_j \bmod qR$, for $\mathbf{R}_j \sim \mathcal{B}_1^{2d\times kd}$ which are appended to the secret key. It then holds that the second block of $\mathbf{A}_{\mathsf{t}_1,\ldots,\mathsf{t}_\omega}$ is equal to $\mathsf{t}\mathbf{G} - \mathbf{A}(\mathbf{R} + \sum_{i\in[\omega]} \mathsf{t}_i \mathbf{R}_i)$ with $\mathsf{t} = \sum_{i\in[\omega]} X^{(i-1)\lfloor n/\omega\rfloor}\mathsf{t}_i \in T_1$. As a result, the effective tag is t and the trapdoor matrix used to sample preimages is $\mathbf{R} + \sum_{i\in[\omega]} \mathsf{t}_i \mathbf{R}_i$ whose spectral norm can be bounded by $(1+\omega\lfloor n/\omega\rfloor)\frac{7}{10}(\sqrt{2nd}+\sqrt{nkd}+6)$.

Note that there are no longer restrictions on the Hamming weight of each tag segment t_i, which means that the tag space is of size $2^{\omega\lfloor n/\omega\rfloor}$, which prevents incidental collisions with overwhelming probability as long as the inputs in are different.

The caveat with having an exponentially large tag space is that the one-more unforgeability proof needs to guess the tag. Fortunately, we can use a confined guessing strategy akin the one from [LLM+16][JRS23, App. G] to avoid this exponential loss in the security proof. More precisely, at the outset of the game, the challenger would sample Q tags $(\mathsf{t}_1^{(i)},\ldots,\mathsf{t}_\omega^{(i)})_{i\in[Q]}$ and reprogram the random oracle queries on in to output the preselected tags. In the branching in Game G_5, we note that if it expects the adversary to use an already queried tag $(\mathsf{t}_1^{(i^+)},\ldots,\mathsf{t}_\omega^{(i^+)})$ in the extracted forgery, the reduction can proceed as for the 3-round case and still end up with a loss term $CQ\varepsilon_{\text{M-SIS}}^{❷}$ (which is independent of the size of the tag space). If it expects a type ❶ one-more forgery, the challenger cannot guess the forgery tag as it will incur an exponential loss. Instead, it samples $i_\ell \hookleftarrow U([Q])$ and $\ell^+ \hookleftarrow U([\omega])$. Then, with probability $1/Q\omega$, the challenger correctly guesses the longest common prefix between the forgery tag $(\mathsf{t}_1^\star,\ldots,\mathsf{t}_\omega^\star)$ and the issued tags. More formally, it expects that

$$\forall j \in [\ell^+ - 1], \mathsf{t}_j^\star = \mathsf{t}_j^{(i_\ell)}, \quad \text{and} \quad \forall i \in [Q], \mathsf{t}_{\ell^+}^\star \neq \mathsf{t}_{\ell^+}^{(i)}.$$

Then, we only have to guess the value of $\mathsf{t}_{\ell^+}^\star$. So the challenger samples $\mathsf{t}_{\ell^+}^+$ uniformly and also sets $\mathsf{t}_j^+ = \mathsf{t}_j^{(i_\ell)}$ for $j < \ell^+$. Using the same hybrid games we progressively replace $\mathbf{B} = \mathbf{A}\mathbf{R}$ by $\mathbf{B} = \mathbf{A}\mathbf{R} + (\sum_{j\in[\ell^+]} \mathsf{t}_j^+ X^{(j-1)\lfloor n/\omega\rfloor})\mathbf{G}$, and finally change $\mathbf{B}_j = \mathbf{A}\mathbf{R}_j \bmod qR$ for $j > \ell^+$. In the end, when extracting the proof from the blind signature, the guess of the ℓ^+ first tag segments is correct with probability $1/(Q\omega 2^{\lfloor n/\omega\rfloor})$. In that case, the second block of the matrix is

$$\begin{aligned}\sum_{i\in[\omega]} \mathsf{t}_i \mathbf{B}_i - \mathbf{B} &= \left(\sum_{j\in[\ell^+]} (\mathsf{t}_j^\star - \mathsf{t}_j^+) X^{(j-1)\lfloor n/\omega\rfloor}\right)\mathbf{G} - \mathbf{A}\mathbf{R} + \sum_{j>\ell^+} \mathsf{t}_j^\star \mathbf{A}\mathbf{R}_j \\ &= \mathbf{A}\left(-\mathbf{R} + \sum_{j>\ell^+} \mathsf{t}_j^\star \mathbf{R}_j\right)\end{aligned}$$

We can therefore re-write the verification equation as $[\mathbf{A}|\mathbf{d}|\mathbf{u}]\mathbf{x}^\star = \mathbf{0} \bmod qR$ with

$$\mathbf{x}^\star = \begin{bmatrix} \mathbf{w}_1 + \left[-\mathbf{R} + \sum_{j>\ell^+} \mathfrak{t}_j^\star \mathbf{R}_j \middle| \mathbf{R}' \right] \mathbf{w}_{23} \\ -\mathcal{H}(m_j) \\ -1 \end{bmatrix}$$

which allows to conclude our security proof as well. Due to the confined guessing, the corresponding loss term is $CQ\omega 2^{\lfloor n/\omega \rfloor} \varepsilon^{❶}_{\text{M-SIS}}$ which can be made polynomial depending on the value of ω. By tweaking the parameter ω, we can thus have a 2-round blind signature while trading off between the security reduction loss and the size of the signer's keys.

Additionally, the relation to be proven with Prove_2 would be different so as to account for the different tag structure. In particular, it would require committing to each $\mathfrak{t}_i$ individually.

Acknowledgments

This work has received a French government support managed by the National Research Agency through the MobiS5 project with reference ANR-18-CE-39-0019-02 MobiS5.

References

ABB20. N. Alkeilani Alkadri, R. El Bansarkhani, and J. Buchmann. BLAZE: Practical Lattice-Based Blind Signatures for Privacy-Preserving Applications. In *Financial Cryptography and Data Security*, 2020.

ACF+15. M. R. Albrecht, C. Cid, J.-C. Faugère, R. Fitzpatrick, and L. Perret. Algebraic Algorithms for LWE Problems. *ACM Commun. Comput. Algebra*, 2015.

AG11. S. Arora and R. Ge. New Algorithms for Learning in Presence of Errors. In *ICALP*, 2011.

AGJ+24. S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois, and O. Sanders. Practical Post-Quantum Signatures for Privacy. In *CCS*, 2024.

AKSY22. S. Agrawal, E. Kirshanova, D. Stehlé, and A. Yadav. Practical, Round-Optimal Lattice-Based Blind Signatures. In *CCS*, 2022.

APS15. M. R. Albrecht, R. Player, and S. Scott. On the Concrete Hardness of Learning With Errors. *J. Math. Cryptol.*, 2015.

Ban93. W. Banaszczyk. New Bounds in Some Transference Theorems in the Geometry of Numbers. *Math. Ann.*, 1993.

BB08. D. Boneh and X. Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. *J. Cryptol.*, 2008.

BCC04. E. F. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *CCS*, 2004.

BCE+20. S. Bouaziz-Ermann, S. Canard, G. Eberhart, G. Kaim, A. Roux-Langlois, and J. Traoré. Lattice-based (Partially) Blind Signature without Restart. *IACR Cryptol. ePrint Arch.*, page 260, 2020.

BJRW23. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the Hardness of Module Learning with Errors with Short Distributions. *J. Cryptol.*, 2023.

BKW03. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant Learning, the Parity Problem, and the Statistical Query Model. *J. ACM*, 2003.

BLL+22. F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)Security of ROS. *J. Cryptol.*, 2022.

BLNS23a. W. Beullens, V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Lattice-Based Blind Signatures: Short, Efficient, and Round-Optimal. In *CCS*, 2023.

BLNS23b. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In *CRYPTO*, 2023.

BMW03. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT*, 2003.

BNPS03. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *J. Cryptol.*, 2003.

BW16. D. Bernhard and B. Warinschi. Cryptographic Voting - A Gentle Introduction. *IACR Cryptol. ePrint Arch.*, page 765, 2016.

CCD+23. J. H. Cheon, H. Choe, J. Devevey, T. Güneysu, D. Hong, M. Krausz, G. Land, M. Möller, D. Stehlé, and M. Yi. HAETAE: Shorter Lattice-Based Fiat-Shamir Signatures. *IACR Cryptol. ePrint Arch.*, page 624, 2023.

Cha82. D. Chaum. Blind Signatures for Untraceable Payments. In *CRYPTO*, 1982.

Cha83. D. Chaum. Blind Signature System. In *CRYPTO*, 1983.

Che13. Y. Chen. *Réduction de Réseau et Sécurité Concrète du Chiffrement Complètement Homomorphe*. PhD thesis, Paris 7, 2013.

CL01. J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT*, 2001.

CL02. J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. In *SCN*, 2002.

CL04. J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, 2004.

CvH91. D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT*, 1991.

DDLL13. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *CRYPTO*, 2013.

DLL+17. L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Cryptol. ePrint Arch.*, page 633, 2017. Version dated from June 27th 2017.

dPK22. R. del Pino and S. Katsumata. A New Framework for More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling. In *CRYPTO*, 2022.

dPLS18. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability. In *CCS*, 2018.

Fis06. M. Fischlin. Round-Optimal Composable Blind Signatures in the Common Reference String Model. In *CRYPTO*, 2006.

GMPW20. N. Genise, D. Micciancio, C. Peikert, and M. Walter. Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography. In *PKC*, 2020.

GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, 2008.

HB23. BIS Innovation Hub and Swiss National Bank. Project Tourbillon: Exploring privacy, security and scalability for CBDCs. https://www.bis.org/publ/othp80.pdf, 2023.

HKLN20. E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-Based Blind Signatures, Revisited. In *CRYPTO*, 2020.

ISO16. ISO/IEC. ISO/IEC 18370-1:2016 Information Technology — Security Techniques — Blind digital signatures. https://www.iso.org/standard/62288.html, 2016.

JLO97. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO*, 1997.

JRS23. C. Jeudy, A. Roux-Langlois, and O. Sanders. Lattice Signature with Efficient Protocols, Application to Anonymous Credentials. In *CRYPTO*, 2023.

LLM$^+$16. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions. In *ASIACRYPT*, 2016.

LN22. V. Lyubashevsky and N. K. Nguyen. BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications. *ASIACRYPT*, 2022.

LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. *CRYPTO*, 2022.

LNPS21. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations. In *ASIACRYPT*, 2021.

LNS21. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments. In *PKC*, 2021.

LS15. A. Langlois and D. Stehlé. Worst-case to Average-case Reductions for Module Lattices. *DCC*, 2015.

Lyu08. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, 2008.

MP12. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, 2012.

MR07. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 2007.

Pei10. C. Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, 2010.

PS00. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptol.*, 2000.

PS16. D. Pointcheval and O. Sanders. Short Randomizable Signatures. In *CT-RSA*, 2016.

Reg09. O. Regev. On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. *J. ACM*, 2009.

Rüc10. M. Rückert. Lattice-Based Blind Signatures. In *ASIACRYPT*, 2010.

Sch89. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO*, 1989.

Wag02. D. A. Wagner. A Generalized Birthday Problem. In *CRYPTO*, 2002.

A Zero-Knowledge Arguments

A.1 Additional Preliminaries

A.1.1 Background on Algebraic Number Theory. We briefly recall some additional details in algebraic number theory, especially on the subring embedding technique used in zero-knowledge arguments, e.g., [LNPS21,AGJ+24].

Subring Embedding. When considering a power-of-two cyclotomic ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power of two, we can naturally benefit from the tower of ring structure and embedding elements of R into smaller degree power-of-two cyclotomic rings. Formally, we let $\widehat{n}$ be another power of two dividing n, and $\widehat{R} = \mathbb{Z}[x]/\langle x^{\widehat{n}} + 1\rangle$. Then, we can map R to $\widehat{R}^{\widehat{k}}$ where $\widehat{k} = n/\widehat{n}$. To do so, we define the subring embedding $\theta : R \to \widehat{R}^{\widehat{k}}$ by

$$\forall a \in R, \theta(a) = \begin{bmatrix} \widehat{a}_0 \\ \vdots \\ \widehat{a}_{\widehat{k}-1} \end{bmatrix} = \begin{bmatrix} \sum_{j\in[0,\widehat{n}-1]} \tau_{j\widehat{k}+0}(a) \cdot x^j \\ \vdots \\ \sum_{j\in[0,\widehat{n}-1]} \tau_{j\widehat{k}+(\widehat{k}-1)}(a) \cdot x^j \end{bmatrix} \in \widehat{R}^{\widehat{k}}$$

If we use $\otimes_R$ to denote the product in R, and $\otimes_{\widehat{R}}$ for the product in $\widehat{R}$ to avoid confusion, it then holds that the inverse embedding is efficiently computable by $\theta^{-1}([\widehat{a}_0, \ldots, \widehat{a}_{\widehat{k}-1}]) = \sum_{i\in[0,\widehat{k}-1]} \widehat{a}_i(x^{\widehat{k}}) \otimes_R x^i$.

Multiplication Matrix. The product in R translates to a matrix-vector multiplication in $\widehat{R}^{\widehat{k}}$ with the subring embedding. More precisely, it holds that $\theta(a \otimes_R b) = M_\theta(a)\theta(b)$ where the matrix-vector product is performed in $\widehat{R}$ and where

$$M_\theta(a) = \begin{bmatrix} \widehat{a}_0 & \widehat{a}_{\widehat{k}-1}x & \ldots & \widehat{a}_1 x \\ \widehat{a}_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \widehat{a}_{\widehat{k}-1}x \\ \widehat{a}_{\widehat{k}-1} & \ldots & \widehat{a}_1 & \widehat{a}_0 \end{bmatrix},$$

where $\widehat{a}_i x = \widehat{a}_i \otimes_{\widehat{R}} x$ is the product in $\widehat{R}$. The subring embedding θ is extended to vectors entrywise and the multiplication map M_θ blockwise to vectors and matrices over R, i.e., for $\mathbf{A} = [a_{i,j}]_{i,j} \in R^{d\times m}$ by $M_\theta(\mathbf{A}) = [M_\theta(a_{i,j})]_{i,j} \in \widehat{R}^{\widehat{k}d\times\widehat{k}m}$.

Coefficient Embedding. The coefficient embedding introduced in Section 2.1 can then be seen as a specific subring embedding where $\widehat{n} = 1$, $\widehat{k} = n$ and $\widehat{R} = \mathbb{Z}[x]/\langle x + 1\rangle = \mathbb{Z}$. The multiplication matrix map M_θ also generalizes M_τ as $\widehat{a}_i \otimes_{\widehat{R}} x = -\widehat{a}_i$ when $\widehat{R}$ is of degree 1.

A.1.2 Dilithium Compression. The zero-knowledge protocol we use embarks the optimization put forth in [LNP22, App. A], namely the compression of the commitments. We thus recall the additional functions and notations that we use for this. These functions aim at extracting low and high-order components of elements of $\mathbb{Z}_q$ (and coefficient-wise for vectors of R_q^d). They are originally presented in Dilithium-G [DLL+17]. We introduce specific notations which are only used in these functions. For an integer a, we call $a' = a \bmod^+ q$ the unique element in $[0, q)$ such that $a' = a \bmod q$. Then, for an even integer α, and an integer a, we let $a' = a \bmod^{\pm} \alpha$ be the unique integer in $(-\alpha/2, \alpha/2]$ such that $a' = a \bmod \alpha$. Note that as opposed to our previous decomposition functions of Section 2.4, $\bmod^{\pm}$ maps to an interval where the lower end is excluded instead of the upper end. We note that in our case, γ is chosen so that it is an even divisor of $q-1$, and such that $\alpha = (q-1)/\gamma$ is also even. More details on these functions can be found in [DLL+17,LNP22].

$\underline{\mathsf{Power2Round}_q(r, D)}$
$r \leftarrow r \bmod^+ q$
$r_L \leftarrow r \bmod^{\pm} 2^D$
return $(r_H, r_L) = ((r - r_L)/2^D, r_L)$

$\underline{\mathsf{UseGHint}_q(y, r, \gamma)}$
$\alpha \leftarrow (q-1)/\gamma$
$r_H \leftarrow \mathsf{HighBits}_q(r, \gamma)$
return $(r_H + y) \bmod^+ \alpha$

$\underline{\mathsf{MakeGHint}_q(z, r, \gamma)}$
$\alpha \leftarrow (q-1)/\gamma$
$r_H \leftarrow \mathsf{HighBits}_q(r, \gamma)$
$v_H \leftarrow \mathsf{HighBits}_q(r + z, \gamma)$
return $(v_H - r_H) \bmod^{\pm} \alpha$

$\underline{\mathsf{Decompose}_q(r, \gamma)}$
$r \leftarrow r \bmod^+ q$
$r_L \leftarrow r \bmod^{\pm} \gamma$
if $r - r_L = q - 1$
  then $(r_H, r_L) \leftarrow (0, r_L - 1)$
else $r_H \leftarrow (r - r_L)/\gamma$
return (r_H, r_L)

$\underline{\mathsf{HighBits}_q(r, \gamma)}$
$(r_H, r_L) \leftarrow \mathsf{Decompose}_q(r, \gamma)$
return r_H

Fig. A.1. Functions for commitment compression.

A.2 Parameter Setting

We use the challenge space from [LNP22], which was also used in other works, e.g., [LN22,BLNS23b,AGJ+24]. More precisely, we let $\rho = \lfloor (2^{2(\lambda+1)/\widehat{n}} - 1)/2 \rfloor$ and η be a positive integer (heuristically determined so that the bound in the definition of $\mathcal{C}$ is verified with probability at least 1/2), and define the challenge space to be

$$\mathcal{C} = \left\{ c \in \widehat{S}_\rho : c^* = c \quad \text{and} \quad \|c^{2k'}\|_1^{1/2k'} \leq \eta \right\}.$$

Because we chose moduli that split into two prime ideals in power-of-two cyclotomic rings, we can rely on [LNP22, Lem. 2.6] to argue that any difference of distinct challenges is in $\widehat{R}_{\widehat{q}}^{\times}$.

We employ the commitment compression technique from [LNP22, App. A]. It involves two parameters γ and D for the low-order bit cuts of $\mathbf{w}$ and $\mathbf{t}_A$ respectively. Both parameters impact the bound of the M-SIS assumption underlying the soundness argument. As such, we proceed as prescribed in [LNP22] by first selecting the largest γ that makes the M-SIS problem still hard, and we then set D to be the largest integer such that $2^{D-1}\rho\widehat{n} < \gamma$. Note that γ is selected so that it is an even divisor of $\widehat{q} - 1$. We thus adjust the modulus factor q_1 so that $q_1 q - 1$ has such a divisor close to the threshold value of γ (threshold above which M-SIS does not meet the security target). Note that this decomposition is recalled in Section A.1.2 and is different from the one presented in Section 2.4 which is used in our blind signature generation procedure.

Once q_1 is set, we define $q_{\min} = \min(q, q_1)$, and set $\ell = \lceil \lambda / 2 \log_2 q_{\min} \rceil$ so that the soundness error term $q_{\min}^{-2\ell}$ is smaller than $2^{-\lambda}$. The parameter ℓ defines the number of parallel repetitions for boosting the soundness, i.e., the number of garbage commitments g_i for the quadratic evaluations. The factor 2 is due to the optimization presented in [LNP22, Sec. 4.4].

A.3 Commitment Opening and Verifiable Encryption Proof

A.3.1 Initial Relation. We now detail how to prove the desired relation in the issuance phase. We start by describing the relation we aim to prove.

After receiving the tag t from the signer, the user sample $\mathbf{r}_1 \hookleftarrow U(\widetilde{S}_{2b_1}^{2d})$, $\mathbf{r}_{23} \hookleftarrow U(\widetilde{S}_{b_2}^{k(d+1)})$ and commits to $\mathcal{H}(m) \in T_1$. Notice that the sampling of $\mathbf{r}_1$ in the scheme is decomposed into first sampling $(\mathbf{r}_{1,L}, \mathbf{r}_{1,H})$ and combining them into $\mathbf{r}_1 = \mathbf{r}_{1,L} + b_1 \mathbf{r}_{1,H}$. By Lemma 2.6, both methods yield the same distribution for $\mathbf{r}_1$. The user also samples $\mathbf{r}_e \hookleftarrow \mathcal{B}_{\eta_e}^{m_e}$ and encrypts $h = \mathcal{H}(m)$ using the verifiable encryption scheme from Section 3.2. The user now wants to prove the following equations. We again insist that the prover only proves knowledge of h but not m itself. As a result, it does not need to prove the hash evaluation circuit.

$$\mathbf{A}\mathbf{r}_1 + [\mathsf{t}\mathbf{G} - \mathbf{B} | \mathbf{A}_3]\mathbf{r}_{23} + \mathbf{d}h = \mathbf{c} \bmod qR \tag{1}$$

$$\begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e + \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \end{bmatrix} h = \begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} \bmod pR \tag{2}$$

$$\|\mathbf{r}_1\|_2^2 \leq B_{r,1}^2 \tag{3}$$

$$\|\mathbf{r}_{23}\|_2^2 \leq B_{r,2}^2 \tag{4}$$

$$\|\mathbf{r}_e\|_2^2 \leq B_{r,e}^2 \tag{5}$$

$$h \in T_1 \tag{6}$$

where $B_{r,1} = 2b_1\sqrt{2nd}$, $B_{r,2} = b_2\sqrt{nk(d+1)}$, and $B_{r,e} = t\sqrt{\eta_e n m_e/2}$. Note that Equation (2) must hold over R_p and not R_q. We explain how to deal with the R_p equation, as well as with the $\mathbf{r}_{1,1}$ element more compactly using the approximate range proof.

A.3.2 Verifiable Encryption with Approximate Range Proof. Because the encryption modulus p is much smaller than all other moduli, it is undesirable to have a proof modulus that has p as factor. To prove the well-formedness of the ciphertexts, we would thus need to prove knowledge of $(\mathbf{r}_e, h, \mathbf{j})$ such that

$$\begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e + \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \end{bmatrix} h + p\mathbf{j}$$

over R. For that, we prove the equation modulo the proof modulus $\widehat{q}$ and argue that all the elements are small enough so that the equation holds over R. As each element of R_p can be seen as an element of $S_{p/2}$, and that $p \ll \widehat{q}$, the argument would go through. In particular, from this equation, we have

$$\begin{aligned} \|\mathbf{j}\|_2 &\leq \frac{1}{p} \left\| \begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \right\|_2 \|\mathbf{r}_e\|_2 + \frac{1}{p} \left\| \begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} \right\|_2 + \lfloor p/2 \rceil \sqrt{n}/p \\ &\leq \frac{\sqrt{n}}{2} (\sqrt{n m_e (d_e + 1)} B_{r,e} + \sqrt{d_e + 1} + 1 + 1/p). \end{aligned}$$

We define $B_{\mathbf{j}} = \frac{\sqrt{n}}{2}(\sqrt{n m_e (d_e + 1)} B_{r,e} + \sqrt{d_e + 1} + 1 + 1/p)$. In the approach described in [LNP22], the authors avoid committing to $\mathbf{j}$ and instead add an approximate range proof in infinity norm to prove that

$$\mathbf{j} = p^{-1} \left(\begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e - \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \end{bmatrix} h \right),$$

has a small infinity norm. In our case, we instead show that this vector is small in Euclidean norm as we already require a Euclidean approximate range proof for other parts of the witness. This avoids computing another high-dimensional challenge matrix $\mathsf{R}^{(e)}$ which was identified in [AGJ+24] as an intensive step when the dimensions are large. Most importantly, it avoids requiring an additional mask $\mathbf{y}^{(e)}$, response $\mathbf{z}^{(e)}$, and rejection sampling step, which would increase the number of repetitions, in turn degrading performances. The Euclidean approximate range proof will give us the desired conclusion without impacting other parts of the proof. This is mostly due to the fact that the constraint will be (roughly) that the proof modulus $\widehat{q}$ should exceed pB_{arp}. But this is already subsumed by other equations which require $\widehat{q} \gtrsim B_{\text{arp}}^2$. We however insist that, besides from the approximate range proof, we do not prove any specific norm on $\mathbf{j}$. As such the norms of $\mathbf{j}$ only determine the correctness of the proof system for the approximate range proof to go through. Indeed, in the extraction, we would extract $\mathbf{r}_e^\star, h^\star$ and define the vector

$$\mathbf{j}^\star = p^{-1} \left(\begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e^\star - \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \end{bmatrix} h^\star \right) \bmod \widehat{q}R,$$

and from the approximate range proof, we would get that $\|\mathbf{j}^\star\|_\infty \leq \|\mathbf{j}^\star\|_2 \leq B_{\text{arp}}$. We can then re-write the definition of $\mathbf{j}^\star$ as

$$\begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \mathbf{r}_e^\star - \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \end{bmatrix} h^\star - p\mathbf{j}^\star = \mathbf{0} \bmod \widehat{q}R$$

Because all the elements are short with respect to $\widehat{q}$, the equation holds over R if $\widehat{q} > p/2 + p\sqrt{n m_e} B_{r,e}/2 + \lfloor p/2 \rfloor + pB_{\text{arp}}$. We thus remove Equation (2) as it will be proven differently as we just described.

A.3.3 Lifting. First, the lifting simply consists in multiplying the mod-q equations by q_1. If one then proves the equation $q_1 a = q_1 b \bmod q_1 q R$, we can deduce that $a = b \bmod qR$. Indeed, it holds that $q_1 a = q_1 b + q_1 q c$ in R, and thus it also holds in the fraction field K where q_1 is invertible. As a result, we get $a = b + qc$ in R which means $a = b \bmod qR$. We thus change Equation (1) into Equation (7) below.

$$q_1 \mathbf{A} \mathbf{r}_1 + q_1 [t\mathbf{G} - \mathbf{B} | \mathbf{A}_3] \mathbf{r}_{23} + q_1 \mathbf{d} h = q_1 \mathbf{c} \bmod \widehat{q} R \tag{7}$$

A.3.4 Embedding. We now embed the relation in $\widehat{R}$ using θ. The idea is to define an equivalent relation over $\widehat{R}$ in terms of the subring embeddings of the secret elements. First of all, we note that the coefficient embedding of $\theta(a) \in \widehat{R}^{\widehat{k}}$ is a simple permutation of that of $a \in R$. As a result, Equations (3), (4), (5) and (6) can be expressed directly in terms of their embedding. It gives Equations (8), (9), (10) and (11).

$$\|\theta(\mathbf{r}_1)\|_2^2 \leq B_{r,1}^2 \tag{8}$$
$$\|\theta(\mathbf{r}_{23})\|_2^2 \leq B_{r,2}^2 \tag{9}$$
$$\|\theta(\mathbf{r}_e)\|_2^2 \leq B_{r,e}^2 \tag{10}$$
$$\theta(h) \in \widehat{T}_1^{\widehat{k}} \tag{11}$$

where $\widehat{k} = n/\widehat{n}$ is the dimension of the subring embedding. Let us now look at the linear equations. We essentially apply θ to each row and rewrite it to be in terms of $\theta(\mathbf{r}_i), \theta(h), \theta(\mathbf{r}_e)$. Using the multiplication matrix map M_θ, we can easily express all the linear terms. We can then substitute Equation (7) by Equation (12).

$$q_1 M_\theta(\mathbf{A}) \theta(\mathbf{r}_1) + q_1 M_\theta([t\mathbf{G} - \mathbf{B} | \mathbf{A}_3]) \theta(\mathbf{r}_{23}) + q_1 M_\theta(\mathbf{d}) \theta(h) = q_1 \theta(\mathbf{c}) \bmod \widehat{q} \widehat{R} \tag{12}$$

We also note that the approximate range proof term for the encryption will need to be updated to be expressed in terms of $\theta(\mathbf{r}_e)$ and $\theta(m)$. More precisely, it will be defined as

$$p^{-1} \left(\theta \left(\begin{bmatrix} \mathsf{ct}_0 \\ \mathsf{ct}_1 \end{bmatrix} \right) - M_\theta \left(\begin{bmatrix} \mathbf{A}_e^T \\ \mathbf{b}_e^T \end{bmatrix} \right) \theta(\mathbf{r}_e) - \begin{bmatrix} \mathbf{0} \\ \lfloor p/2 \rceil \mathbf{I}_{\widehat{k}} \end{bmatrix} \theta(h) \right).$$

A.3.5 Norm Inequalities. To tackle the norm inequalities from Equation (8), we use the four-squares decomposition method employed in [BLNS23b,AGJ+24]. More precisely, we compute four integers $a_{1,0}, a_{1,1}, a_{1,2}, a_{1,3}$ such that $B_{r,1}^2 - \|\theta(\mathbf{r}_1)\|_2^2 = a_{1,0}^2 + a_{1,1}^2 + a_{1,2}^2 + a_{1,3}^2$. We define the ring element $a_1 = a_{1,0} + a_{1,1} x + a_{1,2} x^2 + a_{1,3} x^3 \in \widehat{R}$. It then holds that $\|[\theta(\mathbf{r}_1) | a_1]\|_2^2 = B_{r,1}^2$. Proving the equality (while hiding a_1) will indeed prove the desired inequality. We do the same thing for $\theta(\mathbf{r}_{23})$ and $\theta(\mathbf{r}_e)$ and define the ring element $a_{23}, a_e \in \widehat{R}$

such that $\|[\theta(\mathbf{r}_{23})|a_{23}])\|_2^2 = B_{r,2}^2$, and $\|[\theta(\mathbf{r}_e)|a_e])\|_2^2 = B_{r,e}^2$. We thus replace Equation (8), (9) and (10) by Equations (13), (14) and (15).

$$\left\| \begin{bmatrix} \theta(\mathbf{r}_1) \\ a_1 \end{bmatrix} \right\|_2^2 = B_{r,1}^2 \tag{13}$$

$$\left\| \begin{bmatrix} \theta(\mathbf{r}_{23}) \\ a_{23} \end{bmatrix} \right\|_2^2 = B_{r,2}^2 \tag{14}$$

$$\left\| \begin{bmatrix} \theta(\mathbf{r}_e) \\ a_e \end{bmatrix} \right\|_2^2 = B_{r,e}^2 \tag{15}$$

A.3.6 Bimodal Rejection Sampling. We now transform the relation further in order to support bimodal rejection sampling on the witness vector. Concretely, it means that within the proof we would form $\mathbf{z}_1 = \mathbf{y}_1 + c \cdot \mathfrak{b}\mathbf{s}_1$ for a random bit $\mathfrak{b} \in \{-1, 1\}$. This however affects the relation we are proving and we thus need to adjust it slightly. The way this task is performed in [DDLL13] is to consider an even modulus so that the equation is sign-invariant. Another way, described in [LN22], is to additionally commit to $\mathfrak{b}$. Concretely, the proof would start – after lifting, embedding, etc – by sampling $\mathfrak{b} \hookleftarrow U(\{-1, 1\})$ and considering part of the witness vector in the BDLOP part. The relation to be proven would then be slightly different. The idea is to commit to $\mathfrak{b}\mathbf{s}_1$ instead of $\mathbf{s}_1$ in the Ajtai part. The quadratic terms would not change, but the linear ones would need to be multiplied by $\mathfrak{b}$. Also, we would need to prove that $\mathfrak{b}$ is a sign as described in [LNP22, Sec. 5.1]. The problem is that [LNP22, Sec. 5.1] holds for a prime modulus. They prove that $\mathfrak{b}$ is an integer in $\mathbb{Z}_{\widehat{q}}$ and that $\mathfrak{b}^2 = 1 \bmod \widehat{q}\widehat{R}$. Concluding that $\mathfrak{b} \in \{-1, 1\}$ only holds for $\widehat{q}$ prime. If $\widehat{q} = q_1 q$, the equation proves that $\mathfrak{b} \in \{-1, 1, q_1[q_1^{-1}]_q - q[q^{-1}]_{q_1}, q[q^{-1}]_{q_1} - q_1[q_1^{-1}]_q\}$. To circumvent this issue, we can instead commit to $\mathfrak{b}$ in the Ajtai part and use the approximate norm bound on $\mathbf{s}_1$ to conclude that $\mathfrak{b}$ is small. In general, other conditions on the modulus will ensure that the equation $\mathfrak{b}^2 = 1$ holds over $\mathbb{Z}$, thus concluding that $\mathfrak{b} \in \{-1, 1\}$. We note that it is actually sufficient to discard the two problematic solutions by arguing that they are much larger than the approximate norm bound proven on $\mathfrak{b}$. More precisely, the approximate range proof shows that $\|\mathbf{s}_1\|_2 \lesssim \sqrt{\widehat{q}}$ whereas, in most cases, $\left|q_1[q_1^{-1}]_q - q[q^{-1}]_{q_1}\right| \gg \sqrt{\widehat{q}}$. Combined with $\mathfrak{b}^2 = 1 \bmod \widehat{q}$, it thus proves that $\mathfrak{b}$ is a sign. Let us now translate each equation to its new form.

Equations (13), (14) and (15) are sign-invariant which results in no changes whatsoever when multiplying by $\mathfrak{b}$. We can thus express them directly as

$$\left\| \begin{bmatrix} \mathfrak{b}\theta(\mathbf{r}_1) \\ \mathfrak{b}a_1 \end{bmatrix} \right\|_2^2 = B_{r,1}^2 \tag{16}$$

$$\left\| \begin{bmatrix} \mathfrak{b}\theta(\mathbf{r}_{23}) \\ \mathfrak{b}a_{23} \end{bmatrix} \right\|_2^2 = B_{r,2}^2 \tag{17}$$

$$\left\| \begin{bmatrix} \mathfrak{b}\theta(\mathbf{r}_e) \\ \mathfrak{b}a_e \end{bmatrix} \right\|_2^2 = B_{r,e}^2 \tag{18}$$

Equation (11) would need to be transformed so as to prove that $\tau(\mathfrak{b}\theta(h)) \in \{0, \mathfrak{b}\}^{\widehat{n}\widehat{k}}$. In [LN22], the authors argue that it can be done by proving that $\mathfrak{b}\theta(h) - (\mathfrak{b}-1)/2\mathbf{1}_{\widehat{R}^{\widehat{k}}}$ has binary coefficients. Albeit true, the proof would need to first prove an equation modulo $\widehat{q}$ with a masking term $y_\mathfrak{b} \in \widehat{R}$ which do not a priori verify that $y_\mathfrak{b} - 1$ has even coefficients. It thus seems that one need to resort to $2^{-1} \bmod \widehat{q}$ which is generally extremely large, which means it is going to place further constraints on $\widehat{q}$ for that equation to hold over $\mathbb{Z}$. Instead, we proceed in a slighlty different manner by revisiting [LNP22, Lem. 2.5]. In our case, it suffices to show Equation (19).

$$\langle \tau(\mathfrak{b}\theta(h)) \,,\ \tau(\mathfrak{b}\theta(h) - \mathfrak{b}\mathbf{1}_{\widehat{R}^{\widehat{k}}})\rangle = 0 \bmod \widehat{q}\mathbb{Z}. \tag{19}$$

Let us explain how we can conclude based on the other statements we prove. Since we prove that $\mathfrak{b}$ is a sign, if Equation (19) holds, then we can conclude that $\langle \tau(\mathfrak{b}\theta(h)) \,, \tau(\mathfrak{b}\theta(h)) - \mathfrak{b}\mathbf{1}_{\widehat{n}\widehat{k}}\rangle = 0 \bmod \widehat{q}$. Then, using the approximate norm bound B_{arp} on $\mathfrak{b}\theta(h)$ and that $\mathfrak{b}$ is a sign, we have that the equation holds over $\mathbb{Z}$ if $\widehat{q} > B_{\text{arp}}(B_{\text{arp}} + \sqrt{n})$. Finally, we use the following generalization of [LNP22, Lem. 2.5] to conclude that $\tau(\mathfrak{b}\theta(h)) \in \{0, \mathfrak{b}\}^{\widehat{n}\widehat{k}}$.

Lemma A.1 ([LNP22, Lem. 2.5] generalized). *Let N be a positive integer, $\mathbf{x} \in \mathbb{Z}^N$ and $\mathfrak{b} \in \{-1, 1\}$. Then, $\langle \mathbf{x} \,,\ \mathbf{x} - \mathfrak{b}\mathbf{1}_N\rangle = 0$ if and only if $\mathbf{x} \in \{0, \mathfrak{b}\}^N$.*

Proof. If $\mathbf{x} \in \{0, \mathfrak{b}\}^N$, then we trivially have that $\langle \mathbf{x} \,, \mathbf{x} - \mathfrak{b}\mathbf{1}_N\rangle = 0$. Now assume $\langle \mathbf{x} \,,\ \mathbf{x} - \mathfrak{b}\mathbf{1}_N\rangle = 0$. It holds that $\langle \mathbf{x} \,,\ \mathbf{x} - \mathfrak{b}\mathbf{1}_N\rangle = \sum_{i\in[N]} x_i(x_i - \mathfrak{b})$. Yet, because $\mathfrak{b} \in \{-1, 1\}$, the map $a \in \mathbb{Z} \mapsto a(a - \mathfrak{b})$ is positive. As such, $\langle \mathbf{x} \,,\ \mathbf{x} - \mathfrak{b}\mathbf{1}_N\rangle$ is a zero sum of non-negative terms, which implies that each term is zero. As a result, $x_i \in \{0, \mathfrak{b}\}$ for all $i \in [N]$. ☐

We now change Equation (12) to express it in terms of $(\mathfrak{b}\theta(\mathbf{r}_i), \mathfrak{b}\theta(m), \mathfrak{b})$. Since $\mathfrak{b}$ will be proven to be a sign, and because the equation is purely linear, we can simply multiply it by $\mathfrak{b}$ and get the equivalent equation is

$$\begin{aligned} &q_1 M_\theta(\mathbf{A})\mathfrak{b}\theta(\mathbf{r}_1) + q_1 M_\theta([\mathfrak{t}\mathbf{G} - \mathbf{B}|\mathbf{A}_3])\mathfrak{b}\theta(\mathbf{r}_{23}) + q_1 M_\theta(\mathbf{d})\mathfrak{b}\theta(h) \\ &\quad - \mathfrak{b}q_1\theta(\mathbf{c}) = \mathbf{0} \bmod \widehat{q}\widehat{R} \end{aligned} \tag{20}$$

We note that multiplying the whole equation by $\mathfrak{b}$ does not work for equations featuring quadratic terms because it would make the equation cubic. For the approximate range proof term of encryption, we also need to adjust it. We can virtually see it as if we committed to $\mathfrak{b}\mathbf{j}$ instead of $\mathbf{j}$. It means that the approximate range proof will feature the term

$$p^{-1}\left(\mathfrak{b}\theta\left(\begin{bmatrix}\mathsf{ct}_0\\ \mathsf{ct}_1\end{bmatrix}\right) - M_\theta\left(\begin{bmatrix}\mathbf{A}_e^T\\ \mathbf{b}_e^T\end{bmatrix}\right)\mathfrak{b}\theta(\mathbf{r}_e) - \begin{bmatrix}\mathbf{0}\\ \lfloor p/2\rceil\mathbf{I}_{\widehat{k}}\end{bmatrix}\mathfrak{b}\theta(h)\right) \tag{21}$$

while still allowing us to extract the correct equation from it. Finally, the equations needed to prove that $\mathfrak{b}$ is a sign are

$$\forall i \in [\widehat{n} - 1], \langle \tau(\mathfrak{b}) \,,\ \tau(x^i)\rangle = 0 \bmod \widehat{q}\mathbb{Z} \tag{22}$$

$$\mathfrak{b}^2 - 1 = 0 \bmod \widehat{q}\widehat{R} \tag{23}$$

A.3.7 Requirements on $\widehat{q}$. We note that the majority of the equations we prove must hold over $\mathbb{Z}$ to be meaningful. This is ensured by the approximate range proof subroutine which derives a approximate norm on the witness vector later denoted by $\mathbf{s}_1$ (and the term from Equation (21)) from the norm of a low-dimensional projection of it. Typically, we get that $\|\mathbf{s}_1\|_2^2$ is bounded by B_{arp}^2 with high probability where $B_{\text{arp}}^2 = \frac{2}{13}\lfloor c_{256}^2 \cdot 337\gamma_3^2 B^2 \cdot 256\rfloor$, with $B^2 = B_{r,1}^2 + B_{r,2}^2 + B_{r,e}^2 + n + B_{\mathbf{j}}^2 + 1$. To derive this approximate bound, we rely on [LNP22, Lem. 2.9] which requires

$$\widehat{q} > 41\widehat{n}m_1 B_{\text{arp}} \tag{24}$$

Next, to be able to prove Equations (16), (17) and (18), we need $-\widehat{q} < -B_{r,i}^2$ and $B_{\text{arp}}^2 - B_{r,i}^2 < \widehat{q}$ for $i \in \{1, 2, e\}$.

For Equation (22), we simply need $B_{\text{arp}} < \widehat{q}$. Then, when combined with Equation (23), we either need $-\widehat{q} < -1$ and $B_{\text{arp}}^2 - 1 < \widehat{q}$ for the equation to hold over $\mathbb{Z}$, or simply $\left|q_1[q_1^{-1}]_q - q[q^{-1}]_{q_1}\right| > B_{\text{arp}}$ to discard the two problematic solutions as described above The latter condition is in most cases less restrictive. Once all these equations are true over $\mathbb{Z}$, we can deduce that Equation (19) is also true over $\mathbb{Z}$ if $\widehat{q} > B_{\text{arp}}(B_{\text{arp}} + \sqrt{n})$ as described before. Finally, for as mentioned in Section A.3.2, we need $\widehat{q} > p\sqrt{nm_e}B_{r,e}/2 + pB_{\text{arp}} + p + 1/2$ for the verifiable encryption proof. All things considered, we need $\widehat{q}$ to verify Equation (24) as well Equations (25), (26), (27), (28), (29) and (30) below.

$$\widehat{q} > \max(B_{r,1}^2, B_{r,2}^2, B_{r,e}^2) \tag{25}$$
$$\widehat{q} > B_{\text{arp}}^2 - \min(B_{r,1}^2, B_{r,2}^2, B_{r,e}^2) \tag{26}$$
$$\widehat{q} > B_{\text{arp}} \tag{27}$$
$$\left|q_1[q_1^{-1}]_q - q[q^{-1}]_{q_1}\right| > B_{\text{arp}} \tag{28}$$
$$\widehat{q} > B_{\text{arp}}^2 + B_{\text{arp}}\sqrt{n} \tag{29}$$
$$\widehat{q} > p(\sqrt{nm_e}B_{r,e}/2 + B_{\text{arp}} + 1) + 1/2 \tag{30}$$

In our case, all these conditions will be subsumed by Equation (29).

A.3.8 The Prove_1 protocol. We now describe the protocol to prove Equations (20), (16), (17), (18), (19), (22), (23) (and (21)). To simplify the notations, we define the following quantities.

$$\mathbf{s}_1 = \begin{bmatrix} \mathfrak{b}\theta(\mathbf{r}_1) \\ \mathfrak{b}a_1 \\ \mathfrak{b}\theta(\mathbf{r}_{23}) \\ \mathfrak{b}a_{23} \\ \mathfrak{b}\theta(\mathbf{r}_e) \\ \mathfrak{b}a_e \\ \mathfrak{b}\theta(h) \\ \mathfrak{b} \end{bmatrix}, \quad \text{and} \quad \begin{cases} \mathbf{s}_{1,i} = \mathfrak{b}\theta(\mathbf{r}_i) \ \mathbf{s}'_{1,i} = \begin{bmatrix} \mathfrak{b}\theta(\mathbf{r}_i) \\ \mathfrak{b}a_i \end{bmatrix}, \quad i \in \{1, 23, e\} \\ \mathbf{s}_{1,h} = \mathfrak{b}\theta(h) \ \mathbf{s}_{1,\mathfrak{b}} = \mathfrak{b} \end{cases}$$

The witness dimension is then $m_1 = (2d\widehat{k}+1)+(k(d+1)\widehat{k}+1)+(m_e\widehat{k}+1)+\widehat{k}+1 = \widehat{k}(2d+k(d+1)+m_e+1)+4$. We also define the following public matrices and vectors.

$$\begin{cases} \mathbf{A}_\theta = q_1 M_\theta(\mathbf{A}) \\ \mathbf{B}_\theta = q_1 M_\theta([t\mathbf{G}-\mathbf{B}|\mathbf{A}_3]) \\ \mathbf{D}_\theta = q_1 M_\theta(\mathbf{d}) \end{cases} \qquad \begin{cases} \mathbf{u} = q_1\theta(\mathbf{c}) \\ \mathbf{A}'_e = M_\theta([\mathbf{A}_e|\mathbf{b}_e]^T) \\ \mathbf{I}' = [\mathbf{0}|\lfloor p/2 \rceil \mathbf{I}_{\widehat{k}}]^T \end{cases} \qquad \begin{cases} \mathsf{ct}' = \theta\left(\begin{bmatrix}\mathsf{ct}_0 \\ \mathsf{ct}_1\end{bmatrix}\right) \end{cases}$$

We detail each step of the five rounds needed in the zero-knowledge arguments. As it follows the blueprint from [LNP22,LN22] already described in details in, e.g., [BLNS23b,AGJ+24], we refer to these works for more explanations of each step.

Algorithm A.1: Prove_1.Round-1

1. $\mathbf{s}_{2,1} \hookleftarrow \chi^{m_2-\widehat{d}}$, $\mathbf{s}_{2,2} \hookleftarrow \chi^{\widehat{d}}$
2. $\mathbf{y}_1 \hookleftarrow \mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$
3. $\mathbf{y}_{2,1} \hookleftarrow \mathcal{D}_{\widehat{R}^{m_2-\widehat{d}},\sigma_2}$, $\mathbf{y}_{2,2} \hookleftarrow \mathcal{D}_{\widehat{R}^{\widehat{d}},\sigma_2}$
4. $\mathbf{y}_3 \hookleftarrow \mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$
5. $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0 \wedge \tau_{\widehat{n}/2}(x) = 0\}^\ell)$
6. $\widehat{\mathbf{m}} \leftarrow [\mathbf{y}_3^T|\mathbf{g}^T]^T$
7. $\mathbf{t}_A \leftarrow \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}'_2\mathbf{s}_{2,1} + \mathbf{s}_{2,2} \bmod \widehat{q}\widehat{R}$
8. $\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}'_2\mathbf{y}_{2,1} + \mathbf{y}_{2,2} \bmod \widehat{q}\widehat{R}$
9. $\mathbf{t}_B \leftarrow \mathbf{B}_{yg}\mathbf{s}_{2,1} + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$
10. $(\mathbf{w}_H, \mathbf{w}_L) \leftarrow \mathsf{Decompose}_{\widehat{q}}(\mathbf{w}, \gamma)$
11. $(\mathbf{t}_{A,H}, \mathbf{t}_{A,L}) \leftarrow \mathsf{Power2Round}_{\widehat{q}}(\mathbf{t}_A, D)$

$(\mathsf{R}_0, \mathsf{R}_1) = \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) \in (\{0,1\}^{256\times\widehat{n}(m_1+\widehat{k}d_e+\widehat{k})})^2$ with $\mathsf{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H)$.

Algorithm A.2: Prove_1.Round-2

1. $\mathbf{z}_3^{\mathbb{Z}} \hookleftarrow \tau(\mathbf{y}_3) + \mathsf{R}\begin{bmatrix}\tau(\mathbf{s}_1) \\ \tau(p^{-1}(\mathbf{s}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{s}_{1,e} - \mathbf{I}'\mathbf{s}_{1,h}))\end{bmatrix}$ ▷ $\mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$

$(\gamma_{i,j})_{\substack{i\in[2\ell] \\ j\in[259+\widehat{n}]}} = \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) \in \mathbb{Z}_{\widehat{q}}^{2\ell\times 259+\widehat{n}}$ with $\mathsf{msg}_2 = \mathbf{z}_3^{\mathbb{Z}}$.

Algorithm A.3: Prove_1.Round-3

1. **For** $i \in [2\ell]$ **do**
   $\mathsf{tmp}_i \leftarrow \sum_{j\in[256]}\gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_{j,1}^*\mathbf{s}_1 + \mathsf{r}_{j,2}^*\cdot p^{-1}(\mathbf{s}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{s}_{1,e} - \mathbf{I}'\mathbf{s}_{1,h}) - \mathbf{z}_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{s}'_{1,1}{}^*\mathbf{s}'_{1,1} - B_{r,1}^2) + \gamma_{i,258}(\mathbf{s}'_{1,23}{}^*\mathbf{s}'_{1,23} - B_{r,2}^2) + \gamma_{i,259}(\mathbf{s}'_{1,e}{}^*\mathbf{s}'_{1,e} - B_{r,e}^2) + \gamma_{i,260}\mathbf{s}_{1,h}^*(\mathbf{s}_{1,h} - \mathbf{s}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) - \sum_{j\in[\widehat{n}-1]}\gamma_{i,260+j}\mathbf{s}_{1,\mathfrak{b}}x^{\widehat{n}-j}$
2. **For** $i \in [\ell]$ **do**
   $f_i \leftarrow g_i + 2^{-1}(\mathsf{tmp}_{2i-1} + \mathsf{tmp}_{2i-1}^*) + x^{\widehat{n}/2}\cdot 2^{-1}(\mathsf{tmp}_{2i} + \mathsf{tmp}_{2i}^*) \bmod \widehat{q}\widehat{R}$

$(\mu_i)_i = \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) \in \widehat{R}_{\widehat{q}}^{\ell+d\widehat{k}+1}$ with $\mathsf{msg}_3 = (f_1, \ldots, f_\ell)$.

By developping the expression of f_i, we can express it as follows:

$$\begin{aligned}
f_i = g_i &+ (\gamma_{2i-1,257} + x^{\widehat{n}/2}\gamma_{2i,257})\mathbf{s}'_{1,1}{}^*\mathbf{s}'_{1,1} \\
&+ (\gamma_{2i-1,258} + x^{\widehat{n}/2}\gamma_{2i,258})\mathbf{s}'_{1,23}{}^*\mathbf{s}'_{1,23} \\
&+ (\gamma_{2i-1,259} + x^{\widehat{n}/2}\gamma_{2i,259})\mathbf{s}'_{1,e}{}^*\mathbf{s}'_{1,e} \\
&+ 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{s}^*_{1,h}(\mathbf{s}_{1,h} - \mathbf{s}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) + (\mathbf{s}^*_{1,h} - \mathbf{s}^*_{1,\mathfrak{b}}\mathbf{1}^*_{\widehat{R}^{\widehat{k}}})\mathbf{s}_{1,h}) \\
&+ 2^{-1}(SE_{2i-1} + x^{\widehat{n}/2}SE_{2i})\mathbf{y}_3^* + 2^{-1}(SE^*_{2i-1} + x^{\widehat{n}/2}SE^*_{2i})\mathbf{y}_3 \\
&+ 2^{-1}(SR_{2i-1,1} + x^{\widehat{n}/2}SR_{2i,1})\mathbf{s}_1^* + 2^{-1}(SR^*_{2i-1,1} + x^{\widehat{n}/2}SR^*_{2i,1})\mathbf{s}_1 \\
&+ 2^{-1}(SR_{2i-1,2} + x^{\widehat{n}/2}SR_{2i,2})p^{-1}(\mathbf{s}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{s}_{1,e} - \mathbf{I}'\mathbf{s}_{1,h})^* \\
&\quad + 2^{-1}(SR^*_{2i-1,2} + x^{\widehat{n}/2}SR^*_{2i,2})p^{-1}(\mathbf{s}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{s}_{1,e} - \mathbf{I}'\mathbf{s}_{1,h}) \\
&+ c_i\mathbf{s}^*_{1,\mathfrak{b}} + c'_i\mathbf{s}_{1,\mathfrak{b}} \\
&- d_i
\end{aligned}$$

where

$$SE_{i^\star} = \sum_{j\in[256]} \gamma_{i^\star,j}\mathbf{e}_j, \quad \text{and} \quad SR_{i^\star,l} = \sum_{j\in[256]} \gamma_{i^\star,j}\mathsf{r}_{j,l}, \tag{31}$$

and the polynomials c_i, c'_i and d_i are defined by

$$\begin{aligned}
c_i = 2^{-1}\Bigg(&-\gamma_{2i,260+\widehat{n}/2} + \sum_{j=1}^{\widehat{n}/2-1}(\gamma_{2i-1,260+j} - \gamma_{2i,260+j+\widehat{n}/2})x^j + \gamma_{2i-1,260+\widehat{n}/2}x^{\widehat{n}/2} \\
&+ \sum_{j=\widehat{n}/2+1}^{\widehat{n}-1}(\gamma_{2i-1,260+j} + \gamma_{2i,260+j-\widehat{n}/2})x^j\Bigg) \qquad (32) \\
c'_i = 2^{-1}\Bigg(&\gamma_{2i,260+\widehat{n}/2} + \sum_{j=1}^{\widehat{n}/2-1}(\gamma_{2i,260+\widehat{n}/2-j} - \gamma_{2i-1,260+\widehat{n}-j})x^j - \gamma_{2i-1,260+\widehat{n}/2}x^{\widehat{n}/2} \\
&+ \sum_{j=\widehat{n}/2+1}^{\widehat{n}-1} -(\gamma_{2i-1,260+\widehat{n}-j} + \gamma_{2i,260+3\widehat{n}/2-j})x^j\Bigg) \qquad (33) \\
d_i = \Bigg(&\sum_{j\in[256]}\gamma_{2i-1,j}z^{\mathbb{Z}}_{3,j} + \gamma_{2i-1,257}B^2_{r,1} + \gamma_{2i-1,258}B^2_{r,2} + \gamma_{2i-1,259}B^2_{r,e}\Bigg) \\
&+ x^{\widehat{n}/2}\Bigg(\sum_{j\in[256]}\gamma_{2i,j}z^{\mathbb{Z}}_{3,j} + \gamma_{2i,257}B^2_{r,1} + \gamma_{2i,258}B^2_{r,2} + \gamma_{2i,259}B^2_{r,e}\Bigg) \qquad (34)
\end{aligned}$$

Algorithm A.4: Prove_1.Round-4

1. $\widehat{\mathbf{m}}_y \leftarrow -\mathbf{B}_{yg}\mathbf{y}_{2,1} \bmod \widehat{q}\widehat{R}$
2. $e_0 \leftarrow \sum_{i\in[\ell]}\mu_i\Big((\gamma_{2i-1,257} + x^{\widehat{n}/2}\gamma_{2i,257})\mathbf{y}'_{1,1}{}^*\mathbf{y}'_{1,1} + (\gamma_{2i-1,258} + x^{\widehat{n}/2}\gamma_{2i,258})\mathbf{y}'_{1,23}{}^*\mathbf{y}'_{1,23}$
   $+(\gamma_{2i-1,259} + x^{\widehat{n}/2}\gamma_{2i,259})\mathbf{y}'_{1,e}{}^*\mathbf{y}'_{1,e}$
   $+ 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{y}^*_{1,h}(\mathbf{y}_{1,h} - \mathbf{y}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) + (\mathbf{y}^*_{1,h} - \mathbf{y}^*_{1,\mathfrak{b}}\mathbf{1}^*_{\widehat{R}^{\widehat{k}}})\mathbf{y}_{1,h})\Big)$

$+\mu_{\ell+d\widehat{k}+1}\mathbf{y}_{1,\mathfrak{b}}^2$

3. Compute $SE_1, SR_{1,1}, SR_{1,2}, \ldots, SE_{2\ell}, SR_{2\ell,1}, SR_{2\ell,2}$ and $c_1, c_1' \ldots, c_\ell, c_\ell'$ as in Equations (62), (63) and (64)
4. $e_1 \leftarrow \sum_{i\in[\ell]} \mu_i \Big([\widehat{\mathbf{m}}_y]_{256/\widehat{n}+i}$
$$+2^{-1}[\widehat{\mathbf{m}}_y]^*_{:\frac{256}{\widehat{n}}}(SE_{2i-1} + x^{\widehat{n}/2}SE_{2i}) + 2^{-1}(SE^*_{2i-1} + x^{\widehat{n}/2}SE^*_{2i})[\widehat{\mathbf{m}}_y]_{:\frac{256}{\widehat{n}}}$$
$$+2^{-1}\mathbf{y}_1^*(SR_{2i-1,1} + x^{\widehat{n}/2}SR_{2i,1}) + 2^{-1}(SR^*_{2i-1,1} + x^{\widehat{n}/2}SR^*_{2i,1})\mathbf{y}_1$$
$$+2^{-1}p^{-1}(\mathbf{y}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}_e'\mathbf{y}_{1,e} - \mathbf{I}'\mathbf{y}_{1,h})^*(SR_{2i-1,2} + x^{\widehat{n}/2}SR_{2i,2})$$
$$+2^{-1}p^{-1}(SR^*_{2i-1,2} + x^{\widehat{n}/2}SR^*_{2i,2})(\mathbf{y}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}_e'\mathbf{y}_{1,e} - \mathbf{I}'\mathbf{y}_{1,h})$$
$$+c_i\mathbf{y}^*_{1,\mathfrak{b}} + c_i'\mathbf{y}_{1,\mathfrak{b}}$$
$$+(\gamma_{2i-1,257} + x^{\widehat{n}/2}\gamma_{2i,257})(\mathbf{y}'_{1,1}{}^*\mathbf{s}'_{1,1} + \mathbf{s}'_{1,1}{}^*\mathbf{y}'_{1,1})$$
$$+(\gamma_{2i-1,258} + x^{\widehat{n}/2}\gamma_{2i,258})(\mathbf{y}'_{1,23}{}^*\mathbf{s}'_{1,23} + \mathbf{s}'_{1,23}{}^*\mathbf{y}'_{1,23})$$
$$+(\gamma_{2i-1,259} + x^{\widehat{n}/2}\gamma_{2i,259})(\mathbf{y}^*_{1,e}\mathbf{s}_{1,e} + \mathbf{s}^*_{1,e}\mathbf{y}_{1,e})$$
$$+2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{y}^*_{1,h}(\mathbf{s}_{1,h} - \mathbf{s}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) + \mathbf{s}^*_{1,h}(\mathbf{y}_{1,h} - \mathbf{y}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}})$$
$$+ (\mathbf{y}^*_{1,h} - \mathbf{y}^*_{1,\mathfrak{b}}\mathbf{1}^*_{\widehat{R}^{\widehat{k}}})\mathbf{s}_{1,h} + (\mathbf{s}^*_{1,h} - \mathbf{s}^*_{1,\mathfrak{b}}\mathbf{1}^*_{\widehat{R}^{\widehat{k}}})\mathbf{y}_{1,h})\Big)$$
$$+\sum_{i\in[d\widehat{k}]} \mu_{\ell+i}[\mathbf{A}_\theta\mathbf{y}_{1,1} + \mathbf{B}_\theta\mathbf{y}_{1,23} + \mathbf{D}\mathbf{y}_{1,h} - \mathbf{y}_{1,\mathfrak{b}}\mathbf{u}]_i$$
$$+2\mu_{\ell+d\widehat{k}+1}\mathbf{s}_{1,\mathfrak{b}}\mathbf{y}_{1,\mathfrak{b}}$$

5. $t_0 \leftarrow \mathbf{b}^T\mathbf{y}_{2,1} + e_0 \bmod \widehat{q}\widehat{R}$
6. $t_1 \leftarrow \mathbf{b}^T\mathbf{s}_{2,1} + e_1 \bmod \widehat{q}\widehat{R}$

$c = \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) \in \mathcal{C}$ with $\mathsf{msg}_4 = (t_0, t_1)$.

Algorithm A.5: Prove_1.Round-5

1. $\mathbf{z}_1 \hookleftarrow \mathbf{y}_1 + c\mathbf{s}_1$
2. $\mathbf{z}_{2,1} \hookleftarrow \mathbf{y}_{2,1} + c\mathbf{s}_{2,1}$ and $\mathbf{z}'_{2,2} \leftarrow \mathbf{y}_{2,2} + c\mathbf{s}_{2,2}$
3. Set $\mathbf{z}'_2 \leftarrow \begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}'_{2,2}\end{bmatrix}$, $\mathbf{s}_2 \leftarrow \begin{bmatrix}\mathbf{s}_{2,1}\\ \mathbf{s}_{2,2}\end{bmatrix}$
4. $\mathbf{x} \leftarrow \tau([\mathbf{s}_1^T | (p^{-1}(\mathbf{s}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{s}_{1,e} - \mathbf{I}'\mathbf{s}_{1,h}))^T]^T)$
5. $u_{1,3}, u_2 \hookleftarrow U([0,1))$
6. **if** $u_{1,3} > \frac{\exp\left(\pi\frac{\|c\mathbf{s}_1\|_2^2}{\sigma_1^2} + \pi\frac{\|\mathsf{R}\mathbf{x}\|_2^2}{\sigma_3^2}\right)}{M_1M_3\cdot\cosh\left(2\pi\frac{\langle\tau(\mathbf{z}_1),\ \tau(c\mathbf{s}_1)\rangle}{\sigma_1^2} + 2\pi\frac{\langle\mathbf{z}_3^{\mathbb{Z}},\ \mathsf{R}\mathbf{x}\rangle}{\sigma_3^2}\right)}$, **go to** Algorithm A.1
7. **if** $u_2 > \frac{\exp(\pi\|c\mathbf{s}_2\|_2^2/\sigma_2^2)}{M_2\cdot\cosh(2\pi\langle\tau(\mathbf{z}'_2),\ \tau(c\mathbf{s}_2)\rangle/\sigma_2^2)}$, **go to** Algorithm A.1
8. $\mathbf{z}_{2,2} \leftarrow \mathbf{z}'_{2,2} - c\mathbf{t}_{A,L} - \mathbf{w}_L$
9. **if** $\|\mathbf{z}_{2,1}\|_2^2 + \|\mathbf{z}_{2,2}\|_2^2 > B_{\pi,2}$, **go to** Algorithm A.7
10. $\mathbf{h} \leftarrow \mathsf{MakeGHint}_{\widehat{q}}(\mathbf{z}_{2,2}, \gamma\mathbf{w}_H - \mathbf{z}_{2,2}, \gamma)$

Output: $\pi_2 = (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, f_1, \ldots, f_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$

Note here that as opposed to [AGJ+24], we are not performing rejection sampling on $\mathbf{z}_3^{\mathbb{Z}}$ at round 2 but only at round 5. In addition, we perform rejection sampling on the joint distribution of $(\mathbf{z}_1, \mathbf{z}_3^{\mathbb{Z}})$. The main reason is that we use the same bit $\mathfrak{b}$ for the bimodal rejection sampling, which prevents us from separating the two rejection steps. We now argue that our method leads to the same compact parameters as if we used two different bits for each separate rejection, but with only one single bit $\mathfrak{b}$ to commit to by adjusting the rejection step. If we define $\mathbf{z} = [\tau(\mathbf{z}_1)|\mathbf{z}_3^{\mathbb{Z}}] \in \mathbb{Z}^{\widehat{n}m_1+256}$, it equals $[\tau(\mathbf{y}_1)|\tau(\mathbf{y}_3)]+\mathfrak{b}[\tau(c\widetilde{\mathbf{s}_1})|\mathsf{R}\widetilde{\mathbf{x}}]$ where $\widetilde{\mathbf{x}}$ is the actual secret we set out to hide in the approximate range proof and is such that $\mathfrak{b}\widetilde{\mathbf{x}} = \mathbf{x}$. We also define $\widetilde{\mathbf{s}_1}$ similarly, which is the actual witness to be hidden. For clarity, we define $\mathbf{y} = [\tau(\mathbf{y}_1)|\tau(\mathbf{y}_3)]$ and $\mathbf{v} = [\tau(c\widetilde{\mathbf{s}_1})|\mathsf{R}\mathbf{x}]$. As $\mathfrak{b}$ is uniformly random, the source distribution of $\mathbf{z}$ is $\frac{1}{2}\mathcal{D}_{\mathbb{Z}^{\widehat{n}m_1+256},\sqrt{\mathbf{S}},-\mathbf{v}} + \frac{1}{2}\mathcal{D}_{\mathbb{Z}^{\widehat{n}m_1+256},\sqrt{\mathbf{S}},\mathbf{v}}$, where $\sqrt{\mathbf{S}} = \mathrm{diag}(\sigma_1\mathbf{I}_{\widehat{n}m_1}, \sigma_3\mathbf{I}_{256})$. We then perform rejection sampling to achieve a target distribution for $\mathbf{z}$ of $\mathcal{D}_{\mathbb{Z}^{\widehat{n}m_1+256},\sqrt{\mathbf{S}}}$. For a fixed σ_1, σ_3, the joint rejection rate M must satisfy

$$M \geq \frac{\exp\left(\pi\frac{\|c\widetilde{\mathbf{s}_1}\|_2^2}{\sigma_1^2} + \pi\frac{\|\mathsf{R}\widetilde{\mathbf{x}}\|_2^2}{\sigma_3^2}\right)}{\cosh\left(2\pi\frac{\langle\tau(\mathbf{z}_1),\ \tau(c\widetilde{\mathbf{s}_1})\rangle}{\sigma_1^2} + 2\pi\frac{\langle\mathbf{z}_3^{\mathbb{Z}},\ \mathsf{R}\widetilde{\mathbf{x}})\rangle}{\sigma_3^2}\right)}$$

Because $\cosh \geq 1$, it suffices to bound the numerator. We can do so by setting $\sigma_1 = \sqrt{\pi/\ln M_1}K_1$ and $\sigma_3 = \sqrt{\pi/\ln M_3}K_3$ where K_1 and K_3 are the usual bounds on $\|c\widetilde{\mathbf{s}_1}\|_2$ and $\|\mathsf{R}\widetilde{\mathbf{x}}\|_2$ as in [LNP22]. The rejection rate M would then be M_1M_3, which is the same rate as if we were doing two separate rejections. Overall, the only change comes from adapting the rejection probability. Finally, we note that the rejection probability is an even function of $(\widetilde{\mathbf{s}_1}, \widetilde{\mathbf{x}})$, which is why the specification of Algorithm A.5 uses $(\mathbf{s}_1, \mathbf{x})$ directly instead of the "unsigned pair" $(\widetilde{\mathbf{s}_1}, \widetilde{\mathbf{x}})$.

The only caveat in performing this joint rejection is that we cannot reject $\mathbf{z}_3^{\mathbb{Z}}$ early on in the protocol. As such, we have to perform all the computations of rounds 3 and 4 at each iteration. It can therefore cause very slight slowdowns in the proof generation compared to [AGJ+24] where the reported timings were for a protocol that could abort at round 2.

A.3.9 Verification. We now give the algorithm Verify_1 associated to the proof system. We note that as opposed to the interactive version presented in [LNP22, App. A], our proof is non-interactive. In particular, notice the elements $\mathbf{w}_H$ and t_0 are not part of the proof and thus need to be recovered during verification. The vector $\mathbf{w}_H$ can be recovered using $\mathsf{UseGHint}_{\widehat{q}}$ with the proper inputs depending only on the proof elements and the hint vector $\mathbf{h}$. Once $\mathbf{w}_H$ is computed, we can derive all the challenges from round 1 through 3. We can then recompute t_0 and recover the last challenge c and check it is consistent with the one provided in the proof. Finally, we check all the norms of the responses $\mathbf{z}_1, \mathbf{z}_{2,1}$ (with the recovered $\mathbf{z}_{2,2}$), $\mathbf{z}_3^{\mathbb{Z}}$ as well as $\mathbf{h}$. The expressions of $B_{\pi,1}$ and $B_{\pi,3}$ are determined by the Gaussian tail bound of Lemma 2.3, while $B_{\pi,2}$ (also used in Algorithm A.5) is determined by the Gaussian tail bound on the compression error norm bounds. More precisely, we have

$$\begin{cases} B_{\pi,1} = c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1} \\ B_{\pi,2} = c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2} + \left(\eta 2^{D-1} + \frac{\gamma}{2}\right)\sqrt{\widehat{n}\widehat{d}} \\ B_{\pi,3} = c_{256}\sigma_3\sqrt{256} \end{cases}$$

Algorithm A.6: Verify$_1$

1. $\mathsf{tmp} \leftarrow \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c2^D\mathbf{t}_{A,H} \bmod \widehat{q}\widehat{R}$
2. $\mathbf{w}_H \leftarrow \mathsf{UseGHint}_{\widehat{q}}(\mathbf{h}, \mathsf{tmp}, \gamma)$
3. $\mathbf{z}_{2,2} \leftarrow \gamma\mathbf{w}_H - \mathsf{tmp}$
4. $\mathbf{z}_2 \leftarrow \begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}_{2,2}\end{bmatrix}$
5. $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H))$ and $\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$
6. $(\gamma_{i,j}) \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}})$
7. $(\mu_i) \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}}, (f_1, \ldots, f_\ell))$
8. $\widehat{\mathbf{m}}_z \leftarrow c\mathbf{t}_B - \mathbf{B}_{yg}\mathbf{z}_{2,1} \bmod \widehat{q}\widehat{R}$
9. Compute $SE_1, SR_1, \ldots, SE_{2\ell}, SR_{2\ell}$ and $c_1, c_1, d_1 \ldots, c_\ell, c_\ell', d_\ell$ as in Equations (31), (32), (33) and (34)
10. $t_0 \leftarrow \sum_{i\in[\ell]} \mu_i \Big((\gamma_{2i-1,257} + x^{\widehat{n}/2}\gamma_{2i,257}){\mathbf{z}'_{1,1}}^*\mathbf{z}'_{1,1} + (\gamma_{2i-1,258} + x^{\widehat{n}/2}\gamma_{2i,258}){\mathbf{z}'_{1,23}}^*\mathbf{z}'_{1,23}$
$+(\gamma_{2i-1,259} + x^{\widehat{n}/2}\gamma_{2i,259}){\mathbf{z}'_{1,e}}^*\mathbf{z}'_{1,e}$
$+\, 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{z}_{1,h}^*(\mathbf{z}_{1,h} - \mathbf{z}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}\widehat{k}}) + (\mathbf{z}_{1,h}^* - \mathbf{z}_{1,\mathfrak{b}}^*\mathbf{1}_{\widehat{R}\widehat{k}}^*)\mathbf{z}_{1,h})\Big)$
$+\mu_{\ell+d\widehat{k}+1}\mathbf{z}_{1,\mathfrak{b}}^2$
$+c\Big(\sum_{i\in[\ell]} \mu_i \Big([\widehat{\mathbf{m}}_z]_{256/\widehat{n}+i}$
$+2^{-1}[\widehat{\mathbf{m}}_z]^*_{:\frac{256}{\widehat{n}}}(SE_{2i-1} + x^{\widehat{n}/2}SE_{2i}) + 2^{-1}(SE^*_{2i-1} + x^{\widehat{n}/2}SE^*_{2i})[\widehat{\mathbf{m}}_z]_{:\frac{256}{\widehat{n}}}$
$+2^{-1}\mathbf{z}_1^*(SR_{2i-1,1} + x^{\widehat{n}/2}SR_{2i,1}) + 2^{-1}(SR^*_{2i-1,1} + x^{\widehat{n}/2}SR^*_{2i,1})\mathbf{z}_1$
$+2^{-1}p^{-1}(\mathbf{z}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{z}_{1,e} - \mathbf{I}'\mathbf{z}_{1,h})^*(SR_{2i-1,2} + x^{\widehat{n}/2}SR_{2i,2})$
$+2^{-1}p^{-1}(SR^*_{2i-1,2} + x^{\widehat{n}/2}SR^*_{2i,2})(\mathbf{z}_{1,\mathfrak{b}}\mathsf{ct}' - \mathbf{A}'_e\mathbf{z}_{1,e} - \mathbf{I}'\mathbf{z}_{1,h})$
$+c_i\mathbf{z}^*_{1,\mathfrak{b}} + c'_i\mathbf{z}_{1,\mathfrak{b}}\Big)$
$+\sum_{i\in[d\widehat{k}]} \mu_{\ell+i}[\mathbf{A}_\theta\mathbf{z}_{1,1} + \mathbf{B}_\theta\mathbf{z}_{1,23} + \mathbf{D}\mathbf{z}_{1,h} - \mathbf{z}_{1,\mathfrak{b}}\mathbf{u}]_i\Big)$
$-c^2\Big(\sum_{i\in[\ell]} \mu_i(d_i + f_i) + \mu_{\ell+d\widehat{k}+1}\Big)$
$-c \cdot t_1 + \mathbf{b}^T\mathbf{z}_{2,1} \bmod \widehat{q}\widehat{R}$
11. $c' \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}}, (f_1, \ldots, f_\ell), (t_0, t_1))$

Output: $[\![c = c']\!] \wedge [\![\|\mathbf{z}_1\|_2 \leq B_{\pi,1}]\!] \wedge [\![\|\mathbf{z}_2\|_2 \leq B_{\pi,2}]\!] \wedge [\![\|\mathbf{z}_3^{\mathbb{Z}}\|_2 \leq B_{\pi,3}]\!] \wedge [\![\|\mathbf{h}\|_\infty \leq \frac{\widehat{q}-1}{2\gamma}]\!] \wedge [\![\forall i \in [\ell], \tau_0(f_i) = 0 \wedge \tau_{\widehat{n}/2}(f_i) = 0]\!]$.

A.3.10 Proof Size. We can then compute the proof size as in [AGJ+24] where the Gaussian vector are encoded with rANS, and where here the commitment $\mathbf{t}_A$ is compressed. We also account for the size of the hint vector $\mathbf{h}$ as given in [LNP22]. Overall, the proof size is given by

$$|\pi_2| = \widehat{n}\widehat{d}(\lceil\log_2 \widehat{q}\rceil - D + 2.25) + \left(\frac{256}{\widehat{n}} + 2\ell + 1\right)\lceil\log_2 \widehat{q}\rceil + \widehat{n}\lceil\log_2(2\rho+1)\rceil$$
$$+ \widehat{n}m_1\left(\frac{1}{2} + \log_2\sigma_1\right) + \widehat{n}(m_2 - \widehat{d})\left(\frac{1}{2} + \log_2\sigma_2\right) + 256\left(\frac{1}{2} + \log_2\sigma_3\right).$$

A.3.11 Security. Let us now state the security result for the proof system regarding soundness and zero-knowledge. The zero-knowledge property relies more precisely on the extended M-LWE assumption defined in [LN22], which is generally assumed not significantly easier than M-LWE itself. We refer to [LN22, Sec. 3.1] for more details.

Lemma A.2 ([LNP22,LN22] adapted). *Let M_1, M_2, M_3 be in $(1, \infty)$, and let $\alpha_j = \sqrt{\pi / \ln M_j}$ for $j \in [3]$. Let $B_{\mathbf{s}_1} = \sqrt{B_{r,1}^2 + B_{r,2}^2 + B_{r,e}^2 + n + 1}$ be a bound on the witness and $B = \sqrt{B_{\mathbf{s}_1}^2 + B_{\mathbf{j}}^2}$ be a bound on the vector used in the approximate range proof. We then define $\sigma_1 = \alpha_1 \eta B_{\mathbf{s}_1}$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$, and $\sigma_3 = \alpha_3 \sqrt{337} B$. We take the other parameters, especially q_1, be such that the conditions on $\widehat{q}$ of Section A.3.7 are verified. Then, $(\mathsf{Prove}_1, \mathsf{Verify}_1)$ is knowledge sound with an extractor running in expected polynomial time and soundness error*

$$\varepsilon_{sound}^{(1)} = \frac{2}{|\mathcal{C}|} + q_{\min}^{-\widehat{n}/2} + q_{\min}^{-2\ell} + 2^{-128} + \varepsilon_{\text{M-SIS}}^{sound}$$

and zero-knowledge with loss $\varepsilon_{zk}^{(1)} = \varepsilon_{\text{M-LWE}}^{zk} + \mathsf{negl}(\lambda)$. The term $\varepsilon_{\text{M-SIS}}^{sound}$ is the hardness bound of $\text{M-SIS}_{\widehat{n}, \widehat{d}, m_1 + m_2, \widehat{q}, \beta^{(1)}}$ where

$$\beta^{(1)} = 4\eta \sqrt{4 c_{\widehat{n} m_1}^2 \sigma_1^2 \widehat{n} m_1 + \left(2 c_{\widehat{n} m_2} \sigma_2 \sqrt{\widehat{n} m_2} + (2^D \eta + \gamma) \sqrt{\widehat{n} \widehat{d}}\right)^2},$$

while $\varepsilon_{\text{M-LWE}}^{zk}$ is the hardness bound of $\text{M-LWE}_{\widehat{n}, m_2 - \widehat{d} - \lfloor 256/\widehat{n} \rfloor - \ell - 1, m_2, \widehat{q}, \widehat{\mathcal{B}_1}}$.

A.4 Signature Verification Proof

A.4.1 Initial Relation. After receiving the partial signature $(\mathsf{t}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ from the signer, the user checks that the received elements satisfy the correct constraints and moves on to generating the final blind signature. For that, the user decompose each $\mathbf{v}_i - \mathbf{r}_i$. For $i = 1$, it directly separates $\mathbf{r}_{1,L}$ and $\mathbf{r}_{1,H}$ so as to avoid remaining carries. In particular, it obtains $\mathbf{w}_{i,L}, \mathbf{w}_{i,H}$ that verify $\mathbf{v}_i - \mathbf{r}_i = \mathbf{w}_{i,L} + b_i \mathbf{w}_{i,H}$. In particular, we have $\|\mathbf{w}_{1,H}\|_2 \leq (\|\mathbf{v}_1\|_2 + \|\mathbf{r}_1\|_2 + \|\mathbf{w}_{1,L}\|_2)/b_1 \leq B_1'$ and $\|[\mathbf{w}_{2,H} | \mathbf{w}_{3,H}]\|_2 \leq B_2'$ where

$${B_1'}^2 = \left\lfloor (B_1/b_1 + 3\sqrt{2nd})^2 \right\rfloor, \quad \text{and} \quad {B_2'}^2 = \left\lfloor (B_2/b_2 + 2\sqrt{nk(d+1)})^2 \right\rfloor,$$

The division by b_i allows us to reduce the proof modulus compared to the case without compression. The user now wants to prove the following equations.

$$b_1 \mathbf{A} \mathbf{w}_{1,H} + \mathsf{t} \mathbf{G} \mathbf{w}_{2,L} + b_2 \mathsf{t} \mathbf{G} \mathbf{w}_{2,H} - b_2 \mathbf{B} \mathbf{w}_{2,H} + b_3 \mathbf{A}_3 \mathbf{w}_{3,H} = \mathbf{u}' \bmod qR \tag{35}$$

$$\|\mathbf{w}_{1,H}\|_2^2 \leq {B_1'}^2 \tag{36}$$

$$\|[\mathbf{w}_{2,H} | \mathbf{w}_{3,H}]\|_2^2 \leq {B_2'}^2 \tag{37}$$

$$\|\mathsf{t}\|_2^2 = w \tag{38}$$

$$\mathsf{t} \in T_1 \tag{39}$$

where the $\mathbf{w}_{i,L}$ are public and $\mathbf{u}' = \mathbf{u} + \mathbf{d} \cdot \mathcal{H}(m) - \mathbf{A} \mathbf{w}_{1,L} + \mathbf{B} \mathbf{w}_{2,L} - \mathbf{A}_3 \mathbf{w}_{3,L} \bmod qR$. We follow the same blueprint as in Section A.3, namely by lifting it modulo $\widehat{q} = q q_1$, embedding it into $\widehat{R}$, and changing it so as to perform bimodal rejection sampling. In particular, we encompass the optimizations from [LNP22,LN22].

A.4.2 Lifting. We lift Equation (35) by multiplying by q_1 and get the equivalent Equation (40) below.

$$q_1(b_1\mathbf{A}\mathbf{w}_{1,H}+\mathrm{t}\mathbf{G}\mathbf{w}_{2,L}+b_2\mathrm{t}\mathbf{G}\mathbf{w}_{2,H}-b_2\mathbf{B}\mathbf{w}_{2,H}+b_3\mathbf{A}_3\mathbf{w}_{3,H}) = q_1\mathbf{u}' \bmod \widehat{q}R \quad (40)$$

A.4.3 Embedding. As in Section A.3, the equation on the coefficients of the witness directly translate in the subring as the embedding simply permutes the coefficients. The linear terms are also easily handled using the multiplication matrix map M_θ. Note that the term $q_1\mathbf{t}\mathbf{G}\mathbf{w}_{2,L}$ is also linear as $\mathbf{w}_{2,L}$ is known. In particular it can be expressed as

$$\theta(q_1\mathbf{t}\mathbf{G}\mathbf{w}_{2,L}) = \begin{bmatrix} q_1\theta([\mathbf{G}\mathbf{w}_{2,L}]_1 \cdot \mathrm{t}) \\ \vdots \\ q_1\theta([\mathbf{G}\mathbf{w}_{2,L}]_d \cdot \mathrm{t}) \end{bmatrix} = q_1 M_\theta(\mathbf{w}_{2,L}) \cdot \theta(\mathrm{t}).$$

Finally, the term $q_1 b_2\mathbf{t}\mathbf{G}\mathbf{w}_{2,H}$ is quadratic. It was shown in [AGJ+24] that it can be expressed as

$$\theta(q_1 b_2\mathbf{t}\mathbf{G}\mathbf{w}_{2,H}) = \begin{bmatrix} \theta(\mathrm{t})^T \cdot q_1 b_2 \mathbf{G}_0 \cdot \theta(\mathbf{w}_{2,H}) \\ \vdots \\ \theta(\mathrm{t})^T \cdot q_1 b_2 \mathbf{G}_{d\widehat{k}-1} \cdot \theta(\mathbf{w}_{2,H}) \end{bmatrix},$$

where $\mathbf{G}_i = [\mathbf{0}_{\widehat{k}\times i_1\widehat{k}} | M_\theta(x^{\widehat{k}-1-i_2})^T \mathbf{P} | \mathbf{0}_{\widehat{k}\times(d-i_1-1)\widehat{k}}] M_\theta(\mathbf{G})$ with (i_1, i_2) the quotient and remainder of the Euclidean division of i by $\widehat{k}$ and $\mathbf{P}$ the permutation matrix with ones only on the anti-diagonal. From these transformations, we can substitute Equations (40), (36), (37), (38) and (39) by the equivalent Equations (41), (42), (43), (44) and (45).

$$\begin{aligned} & q_1(b_1 M_\theta(\mathbf{A})\theta(\mathbf{w}_{1,H}) + M_\theta(\mathbf{G}\mathbf{w}_{2,L})\theta(\mathrm{t}) + b_2 \begin{bmatrix} \theta(\mathrm{t})^T\mathbf{G}_0\theta(\mathbf{w}_{2,H}) \\ \vdots \\ \theta(\mathrm{t})^T\mathbf{G}_{d\widehat{k}-1}\theta(\mathbf{w}_{2,H}) \end{bmatrix} \\ & - b_2 M_\theta(\mathbf{B})\theta(\mathbf{w}_{2,H}) + b_3 M_\theta(\mathbf{A}_3)\theta(\mathbf{w}_{3,H})) = q_1\theta(\mathbf{u}') \bmod \widehat{q}\widehat{R}. && (41) \\ & \|\theta(\mathbf{w}_{1,H})\|_2^2 \le {B_1'}^2 && (42) \\ & \|[\theta(\mathbf{w}_{2,H})|\theta(\mathbf{w}_{3,H})]\|_2^2 \le {B_2'}^2 && (43) \\ & \|\theta(\mathrm{t})\|_2^2 = w && (44) \\ & \theta(\mathrm{t}) \in \widehat{T}_1^{\widehat{k}} && (45) \end{aligned}$$

A.4.4 Norm Inequalities. We then turn norm inequalities into equalities using the four-squares decomposition method and additionally committing to this decomposition. We thus replace Equations (42) and (43) by Equations (46) and (47).

$$\left\| \begin{bmatrix} \theta(\mathbf{w}_{1,H}) \\ a_1 \end{bmatrix} \right\|_2^2 = {B'_1}^2 \tag{46}$$

$$\left\| \begin{bmatrix} \theta(\mathbf{w}_{2,H}) \\ \theta(\mathbf{w}_{3,H}) \\ a_2 \end{bmatrix} \right\|_2^2 = {B'_2}^2 \tag{47}$$

$$\tag{48}$$

for $a_i = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + a_{i,3}x^3 \in \widehat{R}$ such that $\|a_1\|_2^2 = {B'_1}^2 - \|\theta(\mathbf{w}_{1,H})\|_2^2$ and $\|a_2\|_2^2 = {B'_2}^2 - \|[\theta(\mathbf{w}_{2,H})|\theta(\mathbf{w}_{3,H})]\|_2^2$.

A.4.5 Bimodal Rejection Sampling. We now use the same methodology as in Section A.3 to change the relation so as to support bimodal rejection sampling on the witness vector. Equations (46), (47) and (44) are sign-invariant which results in no changes whatsoever when multiplying by $\mathfrak{b}$. The membership proof to $\tau^{-1}(\{0, \mathfrak{b}\}^{\widehat{n}\widehat{k}})$ of $\theta(\mathfrak{t})$ is also handled as in Section A.3. We can thus express them directly as

$$\left\| \begin{bmatrix} \mathfrak{b}\theta(\mathbf{w}_{1,H}) \\ \mathfrak{b}a_1 \end{bmatrix} \right\|_2^2 = {B'_1}^2 \tag{49}$$

$$\left\| \begin{bmatrix} \mathfrak{b}\theta(\mathbf{w}_{2,H}) \\ \mathfrak{b}\theta(\mathbf{w}_{3,H}) \\ \mathfrak{b}a_2 \end{bmatrix} \right\|_2^2 = {B'_2}^2 \tag{50}$$

$$\|\mathfrak{b}\theta(\mathfrak{t})\|_2^2 = w \tag{51}$$

$$\langle \tau(\mathfrak{b}\theta(\mathfrak{t}))\,,\ \tau(\mathfrak{b}\theta(\mathfrak{t}) - \mathfrak{b}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) \rangle = 0 \bmod \widehat{q}\mathbb{Z} \tag{52}$$

We just note that as opposed to Section A.3 where we do not prove a norm on the message, here Equation (51) shows the tag has norm $\sqrt{w}$. We can then use this to have a much less restrictive modulus condition for Equation (54) to hold over $\mathbb{Z}$. In particular, using the fact that $\mathfrak{b}$ is proven to be a sign and that $\mathfrak{t}$ has norm $\sqrt{w}$, it only requires $\widehat{q} > w + \sqrt{w\widehat{n}\widehat{k}}$. Then, we change Equation (41) into Equation (53), and add the sign proof for $\mathfrak{b}$

$$q_1(b_1 M_\theta(\mathbf{A})\mathfrak{b} \cdot \mathfrak{b}\theta(\mathbf{w}_{1,H}) + M_\theta(\mathbf{G}\mathbf{w}_{2,L})\mathfrak{b} \cdot \mathfrak{b}\theta(\mathfrak{t}) + b_2 \begin{bmatrix} \mathfrak{b}\theta(\mathfrak{t})^T \mathbf{G}_0 \mathfrak{b}\theta(\mathbf{w}_{2,H}) \\ \vdots \\ \mathfrak{b}\theta(\mathfrak{t})^T \mathbf{G}_{d\widehat{k}-1} \mathfrak{b}\theta(\mathbf{w}_{2,H}) \end{bmatrix}$$

$$- b_2 M_\theta(\mathbf{B})\mathfrak{b} \cdot \mathfrak{b}\theta(\mathbf{w}_{2,H}) + b_3 M_\theta(\mathbf{A}_3)\mathfrak{b} \cdot \mathfrak{b}\theta(\mathbf{w}_{3,H})) = q_1\theta(\mathbf{u}') \bmod \widehat{q}\widehat{R}. \tag{53}$$

$$\forall i \in [\widehat{n} - 1], \langle \tau(\mathfrak{b})\,,\ \tau(x^i) \rangle = 0 \bmod \widehat{q}\mathbb{Z} \tag{54}$$

$$\mathfrak{b}^2 - 1 = 0 \bmod \widehat{q}\widehat{R} \tag{55}$$

A.4.6 Requirements on $\widehat{q}$. The approximate range proof ensures that the witness vector $\mathbf{s}_1$ has norm less than B_{arp} with high probability where $B_{\mathrm{arp}}^2 =$

$\frac{2}{13}\left\lfloor c_{256}^2 \cdot 337\gamma_3^2 B^2 \cdot 256\right\rfloor$, with $B^2 = {B_1'}^2 + {B_2'}^2 + w + 1$. To derive this approximate bound, we rely on [LNP22, Lem. 2.9] which requires condition (56). The norm equations, the sign proof, and the tag membership proof then require that the following conditions are met.

$$\widehat{q} > 41\widehat{n}m_1 B_{\mathrm{arp}} \tag{56}$$
$$\widehat{q} > \max({B_1'}^2, {B_2'}^2, w) \tag{57}$$
$$\widehat{q} > B_{\mathrm{arp}}^2 - \min({B_1'}^2, {B_2'}^2, w) \tag{58}$$
$$\widehat{q} > B_{\mathrm{arp}} \tag{59}$$
$$\left|q_1[q_1^{-1}]_q - q[q^{-1}]_{q_1}\right| > B_{\mathrm{arp}} \tag{60}$$
$$\widehat{q} > w + \sqrt{w\widehat{n}\widehat{k}} \tag{61}$$

In our case, all these conditions will be subsumed by $\widehat{q} \geq B_{\mathrm{arp}}^2$.

A.4.7 The Prove$_2$ protocol. We now describe the protocol to prove Equations (53), (49), (50), (51), (54) and (55). To simplify the notations, we define the following quantities.

$$\mathbf{s}_1 = \begin{bmatrix} \mathfrak{b}\theta(\mathbf{w}_{1,H}) \\ \mathfrak{b}a_1 \\ \mathfrak{b}\theta(\mathbf{w}_{2,H}) \\ \mathfrak{b}\theta(\mathbf{w}_{3,H}) \\ \mathfrak{b}a_2 \\ \mathfrak{b}\theta(\mathfrak{t}) \\ \mathfrak{b} \end{bmatrix}, \quad \text{and} \quad \begin{cases} \mathbf{s}_{1,i} = \mathfrak{b}\theta(\mathbf{w}_{i,H}) \\ \mathbf{s}'_{1,1} = \begin{bmatrix} \mathfrak{b}\theta(\mathbf{w}_{1,H}) \\ \mathfrak{b}a_1 \end{bmatrix} \\ \mathbf{s}'_{1,2} = \begin{bmatrix} \mathfrak{b}\theta(\mathbf{w}_{2,H}) \\ \mathfrak{b}\theta(\mathbf{w}_{3,H}) \\ \mathfrak{b}a_2 \end{bmatrix} \\ \mathbf{s}_{1,t} = \mathfrak{b}\theta(\mathfrak{t}) \\ \mathbf{s}_{1,\mathfrak{b}} = \mathfrak{b} \end{cases}$$

The witness dimension is then $m_1 = (2d\widehat{k}+1) + (k(d+1)\widehat{k}+1) + \widehat{k} + 1 = \widehat{k}(2d+k(d+1)+1)+3$. We also define the following public matrices and vectors.

$$\begin{cases} \mathbf{A}' = q_1 b_1 M_\theta(\mathbf{A}) \\ \mathbf{G}' = q_1 M_\theta(\mathbf{G}\mathbf{w}_{2,L}) \\ \mathbf{B}' = q_1 b_2 M_\theta(\mathbf{B}) \end{cases} \qquad \begin{cases} \mathbf{A}'_3 = q_1 b_3 M_\theta(\mathbf{A}_3) \\ \mathbf{u}'' = q_1\theta(\mathbf{u}') \\ \mathbf{G}'_i = q_1 b_2 \mathbf{G}_i \end{cases}$$

where $\mathbf{G}_i$ is defined before Equation (41). We detail each step of the five rounds needed in the zero-knowledge argument as in the issuance proof of Section A.3.

Algorithm A.7: Prove$_2$.Round-1

1. $\mathbf{s}_{2,1} \hookleftarrow \chi^{m_2-\widehat{d}}$, $\mathbf{s}_{2,2} \hookleftarrow \chi^{\widehat{d}}$
2. $\mathbf{y}_1 \hookleftarrow \mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$
3. $\mathbf{y}_{2,1} \hookleftarrow \mathcal{D}_{\widehat{R}^{m_2-\widehat{d}},\sigma_2}$, $\mathbf{y}_{2,2} \hookleftarrow \mathcal{D}_{\widehat{R}^{\widehat{d}},\sigma_2}$
4. $\mathbf{y}_3 \hookleftarrow \mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$
5. $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0 \wedge \tau_{\widehat{n}/2}(x) = 0\}^\ell)$
6. $\widehat{\mathbf{m}} \leftarrow [\mathbf{y}_3^T|\mathbf{g}^T]^T$
7. $\mathbf{t}_A \leftarrow \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}'_2\mathbf{s}_{2,1} + \mathbf{s}_{2,2} \bmod \widehat{q}\widehat{R}$

8. $\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2'\mathbf{y}_{2,1} + \mathbf{y}_{2,2} \bmod \widehat{q}\widehat{R}$
9. $\mathbf{t}_B \leftarrow \mathbf{B}_{yg}\mathbf{s}_{2,1} + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$
10. $(\mathbf{w}_H, \mathbf{w}_L) \leftarrow \mathsf{Decompose}_{\widehat{q}}(\mathbf{w}, \gamma)$
11. $(\mathbf{t}_{A,H}, \mathbf{t}_{A,L}) \leftarrow \mathsf{Power2Round}_{\widehat{q}}(\mathbf{t}_A, D)$

$(\mathsf{R}_0, \mathsf{R}_1) = \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) \in (\{0,1\}^{256\times\widehat{n}m_1})^2$ with $\mathsf{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H)$.

Algorithm A.8: Prove$_2$.Round-2

1. $\mathbf{z}_3^{\mathbb{Z}} \hookleftarrow \tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1)$ $\qquad \triangleright\ \mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$

$(\gamma_{i,j})_{\substack{i\in[2\ell]\\ j\in[259+\widehat{n}]}} = \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) \in \mathbb{Z}_{\widehat{q}}^{2\ell\times 259+\widehat{n}}$ with $\mathsf{msg}_2 = \mathbf{z}_3^{\mathbb{Z}}$.

Algorithm A.9: Prove$_2$.Round-3

1. **For** $i \in [2\ell]$ **do**
   $\mathsf{tmp}_i \leftarrow \sum_{j\in[256]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - \mathbf{z}_{3,j}^{\mathbb{Z}}) + \sum_{j\in[2]} \gamma_{i,256+j}({\mathbf{s}'_{1,j}}^*\mathbf{s}'_{1,j} - {B'_j}^2)$
   $+\gamma_{i,259}(\mathbf{s}_{1,t}^*\mathbf{s}_{1,t} - w) + \gamma_{i,260}\mathbf{s}_{1,t}^*(\mathbf{s}_{1,t} - \mathbf{s}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}}) - \sum_{j\in[\widehat{n}-1]} \gamma_{i,260+j}\mathbf{s}_{1,\mathfrak{b}}x^{\widehat{n}-j}$
2. **For** $i \in [\ell]$ **do**
   $f_i \leftarrow g_i + 2^{-1}(\mathsf{tmp}_{2i-1} + \mathsf{tmp}_{2i-1}^*) + x^{\widehat{n}/2} \cdot 2^{-1}(\mathsf{tmp}_{2i} + \mathsf{tmp}_{2i}^*) \bmod \widehat{q}\widehat{R}$

$(\mu_i)_{i\in[\ell+d\widehat{k}+1]} = \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) \in \widehat{R}_{\widehat{q}}^{\ell+d\widehat{k}+1}$ with $\mathsf{msg}_3 = (f_1, \ldots, f_\ell)$.

By developping the expression of f_i, we can express it as follows:

$$
\begin{aligned}
f_i = g_i &+ \sum_{j\in[2]} (\gamma_{2i-1,256+j} + x^{\widehat{n}/2}\gamma_{2i,256+j}){\mathbf{s}'_{1,j}}^*\mathbf{s}'_{1,j} \\
&+ ((\gamma_{2i-1,259} + \gamma_{2i-1,260}) + x^{\widehat{n}/2}(\gamma_{2i,259} + \gamma_{2i,260}))\mathbf{s}_{1,t}^*\mathbf{s}_{1,t} \\
&- 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{s}_{1,t}^*\mathbf{s}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}^{\widehat{k}}} + \mathbf{s}_{1,\mathfrak{b}}^*\mathbf{1}_{\widehat{R}^{\widehat{k}}}^*\mathbf{s}_{1,t}) \\
&+ 2^{-1}(SE_{2i-1} + x^{\widehat{n}/2}SE_{2i})\mathbf{y}_3^* + 2^{-1}(SE_{2i-1}^* + x^{\widehat{n}/2}SE_{2i}^*)\mathbf{y}_3 \\
&+ 2^{-1}(SR_{2i-1} + x^{\widehat{n}/2}SR_{2i})\mathbf{s}_1^* + 2^{-1}(SR_{2i-1}^* + x^{\widehat{n}/2}SR_{2i}^*)\mathbf{s}_1 \\
&+ c_i\mathbf{s}_{1,\mathfrak{b}}^* + c_i'\mathbf{s}_{1,\mathfrak{b}} \\
&- d_i
\end{aligned}
$$

where

$$
SE_{i^\star} = \sum_{j\in[256]} \gamma_{i^\star,j}\mathbf{e}_j, \quad \text{and} \quad SR_{i^\star} = \sum_{j\in[256]} \gamma_{i^\star,j}\mathsf{r}_j, \tag{62}
$$

and the polynomials c_i, c'_i and d_i are defined by

$$\begin{aligned} c_i = 2^{-1} \Bigg(& -\gamma_{2i,260+\widehat{n}/2} + \sum_{j=1}^{\widehat{n}/2-1} (\gamma_{2i-1,260+j} - \gamma_{2i,260+j+\widehat{n}/2}) x^j + \gamma_{2i-1,260+\widehat{n}/2} x^{\widehat{n}/2} \\ & + \sum_{j=\widehat{n}/2+1}^{\widehat{n}-1} (\gamma_{2i-1,260+j} + \gamma_{2i,260+j-\widehat{n}/2}) x^j \Bigg) \end{aligned} \tag{63}$$

$$\begin{aligned} c'_i = 2^{-1} \Bigg(& \gamma_{2i,260+\widehat{n}/2} + \sum_{j=1}^{\widehat{n}/2-1} (\gamma_{2i,260+\widehat{n}/2-j} - \gamma_{2i-1,260+\widehat{n}-j}) x^j - \gamma_{2i-1,260+\widehat{n}/2} x^{\widehat{n}/2} \\ & + \sum_{j=\widehat{n}/2+1}^{\widehat{n}-1} -(\gamma_{2i-1,260+\widehat{n}-j} + \gamma_{2i,260+3\widehat{n}/2-j}) x^j \Bigg) \end{aligned} \tag{64}$$

$$\begin{aligned} d_i = & \left(\sum_{j\in[256]} \gamma_{2i-1,j} z^{\mathbb{Z}}_{3,j} + \gamma_{2i-1,259} w + \gamma_{2i-1,257} {B'_1}^2 + \gamma_{2i-1,258} {B'_2}^2 \right) \\ & + x^{\widehat{n}/2} \left(\sum_{j\in[256]} \gamma_{2i,j} z^{\mathbb{Z}}_{3,j} + \gamma_{2i,259} w + \gamma_{2i,257} {B'_1}^2 + \gamma_{2i,258} {B'_2}^2 \right) \end{aligned} \tag{65}$$

Algorithm A.10: Prove$_2$.Round-4

1. $\widehat{\mathbf{m}}_y \leftarrow -\mathbf{B}_{yg}\mathbf{y}_{2,1} \bmod \widehat{q}\widehat{R}$
2. $e_0 \leftarrow \sum_{i\in[\ell]} \mu_i \Big(\sum_{j\in[2]} (\gamma_{2i-1,256+j} + x^{\widehat{n}/2}\gamma_{2i,256+j}) {\mathbf{y}'_{1,j}}^* \mathbf{y}'_{1,j}$
   $+((\gamma_{2i-1,259} + \gamma_{2i-1,260}) + x^{\widehat{n}/2}(\gamma_{2i,259} + \gamma_{2i,260}))\mathbf{y}^*_{1,t}\mathbf{y}_{1,t}$
   $- 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{y}^*_{1,t}\mathbf{y}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}\widehat{k}} + \mathbf{y}^*_{1,\mathfrak{b}}\mathbf{1}^*_{\widehat{R}\widehat{k}}\mathbf{y}_{1,t})\Big)$
   $+ \sum_{i\in[d\widehat{k}]} \mu_{\ell+i} \Big(\mathbf{y}^T_{1,t}\mathbf{G}'_i\mathbf{y}_{1,2} + \mathbf{y}_{1,\mathfrak{b}}([\mathbf{A}'\mathbf{y}_{1,1}]_i - [\mathbf{B}'\mathbf{y}_{1,2}]_i + [\mathbf{A}'_3\mathbf{y}_{1,3}]_i + [\mathbf{G}'\mathbf{y}_{1,t}]_i) \Big)$
   $+\mu_{\ell+d\widehat{k}+1}\mathbf{y}^2_{1,\mathfrak{b}}$
3. Compute $SE_1, SR_1, \ldots, SE_{2\ell}, SR_{2\ell}$ and $c_1, c'_1 \ldots, c_\ell, c'_\ell$ as in Equations (62), (63) and (64)
4. $e_1 \leftarrow \sum_{i\in[\ell]} \mu_i \Big([\widehat{\mathbf{m}}_y]_{256/\widehat{n}+i}$
   $+2^{-1}[\widehat{\mathbf{m}}_y]^*_{:\frac{256}{\widehat{n}}}(SE_{2i-1} + x^{\widehat{n}/2}SE_{2i}) + 2^{-1}(SE^*_{2i-1} + x^{\widehat{n}/2}SE^*_{2i})[\widehat{\mathbf{m}}_y]_{:\frac{256}{\widehat{n}}}$
   $+2^{-1}\mathbf{y}^*_1(SR_{2i-1} + x^{\widehat{n}/2}SR_{2i}) + 2^{-1}(SR^*_{2i-1} + x^{\widehat{n}/2}SR^*_{2i})\mathbf{y}_1$
   $+c_i\mathbf{y}^*_{1,\mathfrak{b}} + c'_i\mathbf{y}_{1,\mathfrak{b}}$
   $+ \sum_{j\in[2]} (\gamma_{2i-1,256+j} + x^{\widehat{n}/2}\gamma_{2i,256+j})({\mathbf{y}'_{1,j}}^*\mathbf{s}'_{1,j} + {\mathbf{s}'_{1,j}}^*\mathbf{y}'_{1,j})$
   $+((\gamma_{2i-1,259} + \gamma_{2i-1,260}) + x^{\widehat{n}/2}(\gamma_{2i,259} + \gamma_{2i,260}))(\mathbf{y}^*_{1,t}\mathbf{s}_{1,t} + \mathbf{s}^*_{1,t}\mathbf{y}_{1,t})$
   $-2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})((\mathbf{y}^*_{1,t}\mathbf{s}_{1,\mathfrak{b}} + \mathbf{s}^*_{1,t}\mathbf{y}_{1,\mathfrak{b}})\mathbf{1}_{\widehat{R}\widehat{k}} + \mathbf{1}^*_{\widehat{R}\widehat{k}}(\mathbf{y}_{1,t}\mathbf{s}^*_{1,\mathfrak{b}} + \mathbf{s}_{1,t}\mathbf{y}^*_{1,\mathfrak{b}}))\Big)$
   $+ \sum_{i\in[d\widehat{k}]} \mu_{\ell+i} \Big(\mathbf{s}^T_{1,t}\mathbf{G}'_i\mathbf{y}_{1,2} + \mathbf{s}_{1,\mathfrak{b}}([\mathbf{A}'\mathbf{y}_{1,1}]_i - [\mathbf{B}'\mathbf{y}_{1,2}]_i + [\mathbf{A}'_3\mathbf{y}_{1,3}]_i + [\mathbf{G}'\mathbf{y}_{1,t}]_i)$
   $+\mathbf{y}^T_{1,t}\mathbf{G}'_i\mathbf{s}_{1,2} + \mathbf{y}_{1,\mathfrak{b}}([\mathbf{A}'\mathbf{s}_{1,1}]_i - [\mathbf{B}'\mathbf{s}_{1,2}]_i + [\mathbf{A}'_3\mathbf{s}_{1,3}]_i + [\mathbf{G}'\mathbf{s}_{1,t}]_i) \Big)$
   $+2\mu_{\ell+d\widehat{k}+1}\mathbf{s}_{1,\mathfrak{b}}\mathbf{y}_{1,\mathfrak{b}}$
5. $t_0 \leftarrow \mathbf{b}^T\mathbf{y}_{2,1} + e_0 \bmod \widehat{q}\widehat{R}$
6. $t_1 \leftarrow \mathbf{b}^T\mathbf{s}_{2,1} + e_1 \bmod \widehat{q}\widehat{R}$

$c = \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) \in \mathcal{C}$ with $\mathsf{msg}_4 = (t_0, t_1)$.

Algorithm A.11: Prove_2.Round-5

1. $\mathbf{z}_1 \hookleftarrow \mathbf{y}_1 + c\mathbf{s}_1$
2. $\mathbf{z}_{2,1} \hookleftarrow \mathbf{y}_{2,1} + c\mathbf{s}_{2,1}$ and $\mathbf{z}'_{2,2} \leftarrow \mathbf{y}_{2,2} + c\mathbf{s}_{2,2}$
3. Set $\mathbf{z}'_2 \leftarrow \begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}'_{2,2}\end{bmatrix}$, $\mathbf{s}_2 \leftarrow \begin{bmatrix}\mathbf{s}_{2,1}\\ \mathbf{s}_{2,2}\end{bmatrix}$
4. $u_{1,3}, u_2 \hookleftarrow U([0,1))$
5. **if** $u_{1,3} > \frac{\exp\left(\pi\frac{\|c\mathbf{s}_1\|_2^2}{\sigma_1^2} + \pi\frac{\|\mathsf{R}\tau(\mathbf{s}_1)\|_2^2}{\sigma_3^2}\right)}{M_1 M_3 \cdot \cosh\left(2\pi\frac{\langle \tau(\mathbf{z}_1),\ \tau(c\mathbf{s}_1)\rangle}{\sigma_1^2} + 2\pi\frac{\langle \mathbf{z}_3^{\mathbb{Z}},\ \mathsf{R}\tau(\mathbf{s}_1)\rangle}{\sigma_3^2}\right)}$, **go to** Algorithm A.7
6. **if** $u_2 > \frac{\exp(\pi\|c\mathbf{s}_2\|_2^2/\sigma_2^2)}{M_2\cdot\cosh(2\pi\langle\tau(\mathbf{z}'_2),\ \tau(c\mathbf{s}_2)\rangle/\sigma_2^2)}$, **go to** Algorithm A.7
7. $\mathbf{z}_{2,2} \leftarrow \mathbf{z}'_{2,2} - c\mathbf{t}_{A,L} - \mathbf{w}_L$
8. **if** $\|\mathbf{z}_{2,1}\|_2^2 + \|\mathbf{z}_{2,2}\|_2^2 > B_{\pi,2}$, **go to** Algorithm A.7
9. $\mathbf{h} \leftarrow \mathsf{MakeGHint}_{\widehat{q}}(\mathbf{z}_{2,2}, \gamma\mathbf{w}_H - \mathbf{z}_{2,2}, \gamma)$

Output: $\pi_2 = (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, f_1, \ldots, f_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$

As is done in the issuance proof of Section A.3, we perform rejection sampling on $\mathbf{z}_3^{\mathbb{Z}}$ conjointly with $\mathbf{z}_1$ in round 5.

A.4.8 Verification. We now give the algorithm Verify_2 associated to the proof system. The expressions of $B_{\pi,1}$ and $B_{\pi,3}$ are determined by the Gaussian tail bound of Lemma 2.3, while $B_{\pi,2}$ (also used in Algorithm A.11) is determined by the Gaussian tail bound on the compression error norm bounds. The expressions are the same as those from Section A.3 but the parameters have different values.

$$\begin{cases} B_{\pi,1} = c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1} \\ B_{\pi,2} = c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2} + \left(\eta 2^{D-1} + \frac{\gamma}{2}\right)\sqrt{\widehat{n}\widehat{d}} \\ B_{\pi,3} = c_{256}\sigma_3\sqrt{256} \end{cases}$$

Algorithm A.12: Verify_2

1. $\mathsf{tmp} \leftarrow \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}'_2\mathbf{z}_{2,1} - c2^D\mathbf{t}_{A,H} \bmod \widehat{q}\widehat{R}$
2. $\mathbf{w}_H \leftarrow \mathsf{UseGHint}_{\widehat{q}}(\mathbf{h}, \mathsf{tmp}, \gamma)$
3. $\mathbf{z}_{2,2} \leftarrow \gamma\mathbf{w}_H - \mathsf{tmp}$
4. $\mathbf{z}_2 \leftarrow \begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}_{2,2}\end{bmatrix}$
5. $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H))$ and $\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$
6. $(\gamma_{i,j}) \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}})$
7. $(\mu_i) \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}}, (f_1, \ldots, f_\ell))$
8. $\widehat{\mathbf{m}}_z \leftarrow c\mathbf{t}_B - \mathbf{B}_{yg}\mathbf{z}_{2,1} \bmod \widehat{q}\widehat{R}$
9. Compute $SE_1, SR_1, \ldots, SE_{2\ell}, SR_{2\ell}$ and $c_1, c_1, d_1 \ldots, c_\ell, c'_\ell, d_\ell$ as in Equations (62), (63), (64) and (65)
10. $t_0 \leftarrow \sum_{i\in[\ell]} \mu_i \Big(\sum_{j\in[2]}(\gamma_{2i-1,256+j} + x^{\widehat{n}/2}\gamma_{2i,256+j})\mathbf{z}'_{1,j}{}^*\mathbf{z}'_{1,j}$
    $+((\gamma_{2i-1,259} + \gamma_{2i-1,260}) + x^{\widehat{n}/2}(\gamma_{2i,259} + \gamma_{2i,260}))\mathbf{z}_{1,t}^*\mathbf{z}_{1,t}$
    $- 2^{-1}(\gamma_{2i-1,260} + x^{\widehat{n}/2}\gamma_{2i,260})(\mathbf{z}_{1,t}^*\mathbf{z}_{1,\mathfrak{b}}\mathbf{1}_{\widehat{R}\widehat{k}} + \mathbf{z}_{1,\mathfrak{b}}^*\mathbf{1}_{\widehat{R}\widehat{k}}^*\mathbf{z}_{1,t})\Big)$
    $+ \sum_{i\in[d\widehat{k}]} \mu_{\ell+i}\left(\mathbf{z}_{1,t}^T\mathbf{G}'_i\mathbf{z}_{1,2} + \mathbf{z}_{1,\mathfrak{b}}([\mathbf{A}'\mathbf{z}_{1,1}]_i - [\mathbf{B}'\mathbf{z}_{1,2}]_i + [\mathbf{A}'_3\mathbf{z}_{1,3}]_i + [\mathbf{G}'\mathbf{z}_{1,t}]_i)\right)$
    $+\mu_{\ell+d\widehat{k}+1}\mathbf{z}_{1,\mathfrak{b}}^2$

$$
\begin{aligned}
&+c \textstyle\sum_{i\in[\ell]} \mu_i \Big([\widehat{\mathbf{m}}_z]_{256/\widehat{n}+i} \\
&\quad +2^{-1}[\widehat{\mathbf{m}}_z]^*_{:\frac{256}{\widehat{n}}} (SE_{2i-1} + x^{\widehat{n}/2} SE_{2i}) + 2^{-1}(SE^*_{2i-1} + x^{\widehat{n}/2} SE^*_{2i})[\widehat{\mathbf{m}}_z]_{:\frac{256}{\widehat{n}}} \\
&\quad +2^{-1}\mathbf{z}_1^*(SR_{2i-1} + x^{\widehat{n}/2} SR_{2i}) + 2^{-1}(SR^*_{2i-1} + x^{\widehat{n}/2} SR^*_{2i})\mathbf{z}_1 \\
&\quad +c_i \mathbf{z}^*_{1,\mathfrak{b}} + c'_i \mathbf{z}_{1,\mathfrak{b}} \Big) \\
&-c^2 \Big(\textstyle\sum_{i\in[\ell]} \mu_i (d_i + f_i) + \sum_{i\in[d\widehat{k}]} \mu_{\ell+i} [\mathbf{u}'']_i + \mu_{\ell+d\widehat{k}+1} \Big) \\
&-ct_1 + \mathbf{b}^T \mathbf{z}_{2,1} \bmod \widehat{q}\widehat{R}
\end{aligned}
$$

11. $c' \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, (\mathbf{t}_{A,H}, \mathbf{t}_B, \mathbf{w}_H), \mathbf{z}_3^{\mathbb{Z}}, (f_1, \ldots, f_\ell), (t_0, t_1))$

Output: $[\![c = c']\!] \wedge [\![\|\mathbf{z}_1\|_2 \leq B_{\pi,1}]\!] \wedge [\![\|\mathbf{z}_2\|_2 \leq B_{\pi,2}]\!] \wedge [\![\|\mathbf{z}_3^{\mathbb{Z}}\|_2 \leq B_{\pi,3}]\!] \wedge [\![\|\mathbf{h}\|_\infty \leq \frac{\widehat{q}-1}{2\gamma}]\!] \wedge [\![\forall i \in [\ell], \tau_0(h_i) = 0 \wedge \tau_{\widehat{n}/2}(h_i) = 0]\!]$.

A.4.9 Proof Size. We can then compute the proof size as in [AGJ+24] where the Gaussian vector are encoded with rANS, and where here the commitment $\mathbf{t}_A$ is compressed. We also account for the size of the hint vector $\mathbf{h}$ as given in [LNP22]. Overall, the proof size is given by

$$
\begin{aligned}
|\pi_2| = \widehat{n}\widehat{d}(\lceil \log_2 \widehat{q} \rceil - D + 2.25) + \left(\frac{256}{\widehat{n}} + 2\ell + 1\right) \lceil \log_2 \widehat{q} \rceil + \widehat{n}\lceil \log_2(2\rho+1) \rceil \\
+ \widehat{n} m_1 \left(\frac{1}{2} + \log_2 \sigma_1\right) + \widehat{n}(m_2 - \widehat{d})\left(\frac{1}{2} + \log_2 \sigma_2\right) + 256\left(\frac{1}{2} + \log_2 \sigma_3\right).
\end{aligned}
$$

A.4.10 Norm Gap after Extraction. Recall that the bound proven on the $\mathbf{w}_{i,H}$ was a worst-case bound B'_i (in particular independent on $\mathbf{w}_{i,L}$). As a result, the extracted solution, when recombined with the lower part, will verify $\|\mathbf{w}_1^\star\|_2 \leq b_1 B'_1 + b_1\sqrt{2nd}$, and $\|[\mathbf{w}_2^\star | \mathbf{w}_3^\star]\|_2 \leq b_2 B'_2 + b_2\sqrt{nk(d+1)}$. This small additive gap is then taken into account when deriving the M-SIS bounds in the one-more unforgeability.

A.4.11 Security. Let us now state the security result for the proof system regarding soundness and zero-knowledge just like for the issuance proof system. We note that we used the same notations for the parameters of the two proof systems but the actual value are likely different ($\widehat{d}, m_1, m_2, q_1$, etc.).

Lemma A.3 ([LNP22,LN22] adapted). *Let M_1, M_2, M_3 be in $(1, \infty)$, and let $\alpha_j = \sqrt{\pi/\ln M_j}$ for $j \in [3]$. Let $B_{\mathbf{s}_1} = \sqrt{B_1'^2 + B_2'^2 + w + 1}$ be a bound on the witness. We then define $\sigma_1 = \alpha_1 \eta B_{\mathbf{s}_1}$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$, and $\sigma_3 = \alpha_3 \sqrt{337} B_{\mathbf{s}_1}$. We take the other parameters, especially q_1, be such that the conditions on $\widehat{q}$ of Section A.4.6 are verified. Then, $(\mathsf{Prove}_2, \mathsf{Verify}_2)$ is knowledge sound with an extractor running in expected polynomial time and soundness error*

$$
\varepsilon^{(1)}_{sound} = \frac{2}{|\mathcal{C}|} + q_{\min}^{-\widehat{n}/2} + q_{\min}^{-2\ell} + 2^{-128} + \varepsilon^{sound}_{\text{M-SIS}}
$$

and zero-knowledge with loss $\varepsilon_{zk}^{(1)} = \varepsilon_{\text{M-LWE}}^{zk} + \mathsf{negl}(\lambda)$. *The term* $\varepsilon_{\text{M-SIS}}^{sound}$ *is the hardness bound of* $\text{M-SIS}_{\widehat{n},\widehat{d},m_1+m_2,\widehat{q},\beta^{(2)}}$ *where*

$$\beta^{(2)} = 4\eta\sqrt{4c_{\widehat{n}m_1}^2\sigma_1^2\widehat{n}m_1 + \left(2c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2} + (2^D\eta+\gamma)\sqrt{\widehat{n}\widehat{d}}\right)^2}$$

while $\varepsilon_{\text{M-LWE}}^{zk}$ *is the hardness bound of* $\text{M-LWE}_{\widehat{n},m_2-\widehat{d}-\lfloor 256/\widehat{n}\rfloor-\ell-1,m_2,\widehat{q},\widehat{\mathcal{B}_1}}$.

B Parameters

Symbol	Description	Value
	Signature Parameters	
λ	Security parameter	128
n	Signature ring degree	256
d	Module rank	5
q	Modulus	$8388581 \approx 2^{23}$
k	Gadget length	3
b	Gadget base	204
b_1	First decomposition base	512
b_2	Second decomposition base	8
ε	Smoothing loss for samplers	2^{-40}
s_1	Top preimage sampling width	111520.358
s_2	Bottom preimage sampling width	1156.135
w	Hamming weight of tags	5
κ	Number of splitting factors of q	2
Q	Maximal number of signature queries	2^{32}
α_1, α_2	Rejection sampling slack	2.13
M_1, M_2	Rejection sampling repetition rate	2
B_1	First verification bound	2687499.37
B_2	Second verification bound	35802.64
B_1'	First proof bound	5400.81
B_2'	Second proof bound	4611.09
	Security Estimates	
$\text{BKZ}^{❶}$	Required BKZ blocksize for $\text{M-SIS}^{❶}$	641
$\text{BKZ}^{❷}$	Required BKZ blocksize for $\text{M-SIS}^{❷}$	568
$\text{BKZ}^{(1)}$	Required BKZ blocksize for $\text{M-LWE}^{(1)}$ (key)	484
$\text{BKZ}^{(2)}$	Required BKZ blocksize for $\text{M-LWE}^{(2)}$ (hiding cmt.)	448
$\varepsilon^{❶}_{\text{M-SIS}}$	Hardness bound for $\text{M-SIS}^{❶}$	$2^{-203.5}$
$\varepsilon^{❷}_{\text{M-SIS}}$	Hardness bound for $\text{M-SIS}^{❷}$	$2^{-182.2}$
$\varepsilon^{(1)}_{\text{M-LWE}}$	Hardness bound for $\text{M-LWE}^{(1)}$	$2^{-157.7}$
$\varepsilon^{(2)}_{\text{M-SIS}}$	Hardness bound for $\text{M-LWE}^{(2)}$	$2^{-147.2}$
	Efficiency Estimates	
$\lvert\mathsf{pk}\rvert$	Size of public key ($\mathbf{B}$, rest generated from seed)	53.94 KB
$\lvert\mathsf{sk}\rvert$	Size of secret key ($\mathbf{R}$)	9.38 KB
$\lvert\mathsf{sig}\rvert$	Size of partial signature ($\mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3$)	8.70 KB

Table B.1. Suggested parameter set for the blind signature (signature).

Symbol	Description	Value
	Signature Parameters	
λ	Security parameter	128
n	Signature ring degree	256
d_e	Module rank	3
p	Modulus	$4993 \approx 2^{12.3}$
m_e	Number of samples	7
η_e	Binomial parameter	1
$B_{r,e}$	Binomial tail bound	34.42
$p/B_{r,e}^2$	Decryption correctness gap	4.21
	Security Estimates	
$\mathrm{BKZ}^{(e)}$	Required BKZ blocksize for M-LWE$^{(e)}$ (IND-CPA)	503
$\varepsilon_{\text{M-LWE}}^{(e)}$	Hardness bound for M-LWE$^{(e)}$	$2^{-163.2}$
	Efficiency Estimates	
$\lvert\mathsf{ct}\rvert$	Size of ciphertext	1.63 KB

Table B.2. Suggested parameter set for the blind signature (encryption to the sky).

Symbol	Description	Value
Proof System Parameters		
λ	Security parameter	128
$\widehat{n}$	Proof system ring degree	64
$\widehat{k}$	Subring embedding dimension	4
$\widehat{d}$	Module rank	22
q_1	Modulus factor	$17179868957 \approx 2^{34}$
$q_{\min}$	Smallest modulus factor	8388581
$\widehat{q}$	Proof system modulus (qq_1)	144114722315180017
ℓ	Soundness amplification dimension	3
m_1	Witness dimension	148
m_2	Dimension of ABDLOP randomness	69
χ	Distribution of ABDLOP randomness	$\mathcal{B}_1$
ρ	Infinity norm of challenges	8
η	Manhattan-like norm of challenges	93
$(\alpha_1, \alpha_2, \alpha_3)$	Rejection sampling slacks	$(2.13, 2.13, 2.13)$
(M_1, M_2, M_3)	Rejection sampling repetition rates	$(2, 2, 2)$
σ_1	First rejection sampling width	10258622.895
σ_2	Second rejection sampling width	13157.086
σ_3	Third rejection sampling width	2220943.590
γ	Mask commitment compression parameter	603990638
D	Witness commitment compression parameter	21
Security Estimates		
$\mathrm{BKZ}^{\mathrm{sound}}_{\text{M-SIS}}$	Required BKZ blocksize for M-SIS, Lem. A.2	389
$\mathrm{BKZ}^{\mathrm{zk}}_{\text{M-LWE}}$	Required BKZ blocksize for M-LWE, Lem. A.2	280
$\varepsilon^{\mathrm{sound}}_{\text{M-SIS}}$	Hardness bound for M-SIS, Lem. A.2	$2^{-129.9}$
$\varepsilon^{\mathrm{zk}}_{\text{M-LWE}}$	Hardness bound for M-LWE, Lem. A.2	$2^{-127.3}$
$\varepsilon^{(1)}_{\mathrm{sound}}$	Soundness error	$2^{-128.2}$
$\varepsilon^{(1)}_{\mathrm{zk}}$	Zero-Knowledge security bound	$2^{-127.3}$
Efficiency Estimates		
$\lvert\pi_1\rvert$	Proof size	44.90 KB

Table B.3. Suggested parameter set for the blind signature (Prove_1).

Symbol	Description	Value
	Proof System Parameters	
λ	Security parameter	128
$\widehat{n}$	Proof system ring degree	64
$\widehat{k}$	Subring embedding dimension	4
$\widehat{d}$	Module rank	22
q_1	Modulus factor	$268435157 \approx 2^{28}$
$q_{\min}$	Smallest modulus factor	8388581
$\widehat{q}$	Proof system modulus (qq_1)	2251790057742217
ℓ	Soundness amplification dimension	3
m_1	Witness dimension	119
m_2	Dimension of ABDLOP randomness	65
χ	Distribution of ABDLOP randomness	$\mathcal{B}_1$
ρ	Infinity norm of challenges	8
η	Manhattan-like norm of challenges	93
$(\alpha_1, \alpha_2, \alpha_3)$	Rejection sampling slacks	$(2.13, 2.13, 2.13)$
(M_1, M_2, M_3)	Rejection sampling repetition rates	$(2, 2, 2)$
σ_1	First rejection sampling width	1406027.473
σ_2	Second rejection sampling width	12770.028
σ_3	Third rejection sampling width	277540.144
γ	Mask commitment compression parameter	146557902
D	Witness commitment compression parameter	19
	Security Estimates	
$\mathrm{BKZ}^{\text{sound}}_{\text{M-SIS}}$	Required BKZ blocksize for M-SIS, Lem. A.3	380
$\mathrm{BKZ}^{\text{zk}}_{\text{M-LWE}}$	Required BKZ blocksize for M-LWE, Lem. A.3	379
$\varepsilon^{\text{sound}}_{\text{M-SIS}}$	Hardness bound for M-SIS, Lem. A.3	$2^{-127.3}$
$\varepsilon^{\text{zk}}_{\text{M-LWE}}$	Hardness bound for M-LWE, Lem. A.3	2^{-127}
$\varepsilon^{(2)}_{\text{sound}}$	Soundness error	$2^{-126.9}$
$\varepsilon^{(2)}_{\text{zk}}$	Zero-Knowledge security bound	2^{-127}
	Efficiency Estimates	
$\lvert\pi_2\rvert$	Proof size	35.09 KB

Table B.4. Suggested parameter set for the blind signature (Prove_2).