Lattice Signature with Efficient Protocols, Application to Anonymous Credentials

Corentin Jeudy[1,2], Adeline Roux-Langlois[3], and Olivier Sanders[1]

corentin.jeudy@orange.com, adeline.roux-langlois@cnrs.fr,
olivier.sanders@orange.com

[1] Orange Labs, Applied Crypto Group, Cesson-Sévigné, France
[2] Univ Rennes, CNRS, IRISA, Rennes, France
[3] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract. Digital signature is an essential primitive in cryptography, which can be used as the digital analogue of handwritten signatures but also as a building block for more complex systems. In the latter case, signatures with specific features are needed, so as to smoothly interact with the other components of the systems, such as zero-knowledge proofs. This has given rise to so-called *signatures with efficient protocols*, a versatile tool that has been used in countless applications. Designing such signatures is however quite difficult, in particular if one wishes to withstand quantum computing. We are indeed aware of only one post-quantum construction, proposed by Libert et al. at Asiacrypt'16, yielding very large signatures and proofs.

In this paper, we propose a new construction that can be instantiated in both standard lattices and structured ones, resulting in each case in dramatic performance improvements. In particular, the size of a proof of message-signature possession, which is one of the main metrics for such schemes, can be brought down to less than 650 KB. As our construction retains all the features expected from signatures with efficient protocols, it can be used as a drop-in replacement in all systems using them, which mechanically improves their own performance, and has thus a direct impact on many applications. It can also be used to easily design new privacy-preserving mechanisms. As an example, we provide the first lattice-based anonymous credentials system.

Keywords: Lattice-Based Cryptography · Signature · Efficient Protocols · Privacy · Anonymous Credentials

1 Introduction

Electronic authentication massively relies on digital signatures, a cryptographic primitive that can be traced back to the Diffie-Hellman seminal paper [DH76]. The strong point of digital signatures is that they act in the digital world in the same way as handwritten signatures do in the real world: they add a short element S to some data m attesting that m has been validated by the signer and that it has not been modified afterwards. By emulating handwritten signatures, they represent the perfect electronic counterpart and are indeed ubiquitous today.

However, for several decades, cryptographers have questioned this hegemony in some situations as these signatures may give rise to many privacy issues. Typically, presentation of the same certificate[4] S each time m needs to be authenticated allows tracing S and hence its owner. Moreover, if m is a set of elements m_i, then verification of S requires knowledge of all these elements even if they are irrelevant for the current authentication.

For example, let us consider the classical use-case of age control (e.g., to check that a customer is an adult) where some customer owns a digital certificate (embedded in some ID document) authenticating his attributes (name, birthdate, address, etc). With standard digital signature, this customer has no other choice than providing the full set of attributes to the controller as they are required to run the verification algorithm. This is clearly a significant privacy issue but here one could argue that the situation already occurs in the real world: it is indeed quite common to present an ID document displaying many personal information to a cashier that needs to control your age.

This apparent paradox epitomizes the differences between the real world and the digital one. In the former, it is natural to assume that the cashier will not memorize all the information contained in the document for further commercial exploitation or identity theft. This does not hold true in the digital world where the users definitely lose control of their data as soon as they reveal them and it is very likely that the same customer will be much more reluctant to provide the same information to a website that needs to verify that he is an adult.

1.1 Related Works

Since the problems of the two worlds are different it is actually logical that standard digital signatures are not best suited for all use-cases. In particular, the fact that electronic data can no longer be controlled once they are revealed calls for solutions disclosing as few information as possible during authentication. This has given rise to countless advanced cryptographic primitives, tailored to very specific use-cases, such as anonymous credentials [Cha85,CL01,FHS19], group signatures [CvH91,BSZ05], Direct Anonymous Attestations (DAA) [BCC04], EPID [BL07], etc. Far from simply being theoretical constructions, some of them have been included in standards (e.g., [ISO13a,ISO13b]) and even embedded in billions of devices (e.g., [TCG15,Int16]).

Surprisingly, the diversity of use-cases addressed by these privacy-preserving authentication mechanisms contrasts with the very few mathematical settings allowing efficient designs. A closer look at these standards indeed shows that all of them make use of RSA moduli or cyclic groups and thus cannot withstand the power of quantum computing. The emerging success of such systems is thus

[4] All along this paper, the words *signature* and *certificate* will be used interchangeably.

based on foundations that will crumble as soon as a sufficiently powerful quantum computer appears.

This unsatisfying state of affairs clearly calls for the design of post-quantum alternatives to such systems. However, when we look at the cryptographic literature on this topic, it is striking to see that the existing post-quantum solutions are not only much less efficient than their classical[5] counterparts but also extremely rare. Typically, we are not aware of any explicit post-quantum anonymous credentials system. Even when we consider popular primitives such as group signatures, we note that the most efficient solutions [dPLS18,LNPS21] depart from the traditional model [BSZ05] as they do not achieve non-frameability, a property implying that the certificate issuer does not know users' secret keys and that is thus incompatible with their construction. Although this might seem to be a minor restriction for group signatures, this has very important consequences on their industrial variants such as DAA and EPID. Indeed, for the latter, the knowledge of the users' secret keys allows one to break anonymity, which makes the whole construction totally pointless.

To understand the contrasting situations of classical constructions and post-quantum ones in the area of privacy-preserving authentication mechanisms, it is important to recall that all of them require, at some point, to prove knowledge of a signature on some (potentially secret) attributes. For example, in an anonymous credential system, the user generally receives a signature on their attributes and some secret key at the time of issuance. To show their credentials they then reveal the requested attributes and prove knowledge of the signature, the hidden attributes and the secret key so as to remain anonymous. In non-frameable group signatures, DAA or EPID schemes, the user first receives a certificate C on a secret key s and then generates their own signatures by including a zero-knowledge proof that C is valid on s. Of course, the resulting signatures also contains additional elements that define the specificity of each primitive but the point is that the common core is this proof of knowledge which essentially needs two kinds of building blocks: a "signature scheme with efficient protocols" as coined by Camenisch and Lysyanskaya [CL02] and an associated zero-knowledge (ZK) proof system.

The latter notion is well-known and has seen several advances over the past few years, in particular in the lattice setting, e.g., [BLS19,YAZ$^+$19,LNP22]. The former notion is rather informal but it usually refers to a digital signature scheme with some specific features such as the ability to sign committed (hidden) messages and to prove knowledge of a signature on such messages. This places some restrictions on the design of the signature scheme as it for example proscribes hash functions and hence most popular paradigms such as Hash-and-Sign and Fiat-Shamir. Yet, several extremely efficient constructions from number theoretic assumptions exist, in particular in bilinear (pairing) environments [CL04,BB08,PS16]. They constitute a very powerful and simple-to-use building block which explains the countless applications using them.

[5] In this paper, we use "classical" to denote cryptographic constructions that rely on computational assumptions broken by quantum algorithms.

This situation stands in sharp contrast with the one of post-quantum cryptography where we are aware of only one lattice-based construction [LLM+16] with such features. Moreover the latter was designed with Stern's proof of knowledge in mind and thus does not leverage the recent advances in the area of lattice-based zero-knowledge proofs. The original paper only provides asymptotic estimation but our thorough analysis (deferred in Appendix G) shows that, even with the recent ZK protocol from [YAZ+19], a proof of knowledge of a signature is still, at best, 670 MB large, which is far too high for practical applications. This leaves designers of privacy-preserving systems with no other solution than constructing the whole system from scratch, which requires skills in many different areas and thus limits the number of contributions.

1.2 Our Contributions

The goal of our paper is to propose the lattice counterpart of [CL04,BB08,PS16], that is, a signature scheme with efficient protocols that is specifically designed to smoothly and efficiently interact with the most recent lattice-based zero-knowledge proof systems. More precisely, we provide a lattice-based signature scheme for which we can (1) obtain signatures on potentially hidden (in a commitment) messages, and (2) prove in zero-knowledge the possession of a message-signature pair. Compared to the only such construction [LLM+16], our scheme is not only much more efficient but also transposes well to an algebraically structured setting which leads to further performance improvements, as summarized in Table 1.1.

Our natural starting point is [LLM+16] which consists in a Boyen signature [Boy10] on a randomly chosen tag $\tau \in \{0,1\}^\ell$ and for a syndrome shifted by the binary decomposition of the commitment $\mathbf{c} = \mathbf{D}_0\mathbf{r} + \mathbf{D}_1\mathbf{m}$ to a binary message $\mathbf{m}$, the commitment scheme being implicit in [Ajt96]. At first sight, this scheme perfectly fits the recent zero-knowledge proof system proposed by Yang et al. [YAZ+19] but yet leads to an extremely large proof of knowledge as explained above (a thorough complexity analysis is provided in Appendix F.3 and Table G.1). We then undertake a complete overhaul of this scheme, pointing out at the same time the reasons of such a high complexity.

The main novelty is that we adopt a much more global approach as we look simultaneously at the three components of such systems, namely the commitment scheme (necessary to obtain signature on hidden messages), the signature scheme and the zero-knowledge proof systems, and the possible synergies. We, in particular, emphasize that the design choices we made for each component were not driven by the will to improve the latter individually but rather by their impact on the whole system. Typically, some of the modifications we introduce in the signature scheme itself has almost no impact on its complexity but yet results in very significant gains when it comes to proving knowledge of a signature. More generally, our approach leads to a series of contributions that we regroup in three main parts.

The signature scheme. One of the first consequences of having to sign committed messages is that the signature must now include the randomness added to the commitment by the signer. In [LLM$^+$16], this randomness has the same dimension as the one of the Boyen signature but a much larger width (see Table G.1) and thus represents the largest part of the signature. This is amplified by the proof of knowledge, which explains in part the high complexity of the latter. One of the reasons of such a large width is that the security proof requires to embed a hidden relation in the matrix $\mathbf{D}$ that is applied to the binary decomposition of the Ajtai commitment $\mathbf{c}$. More precisely, it defines $\mathbf{D} = \mathbf{AU}$ for the matrix $\mathbf{A}$ from the Boyen public key and some short matrix $\mathbf{U}$. This (along with other design choices discussed below) deteriorates the quality of the SIS solution extracted during the security proof and thus leads to large parameters.

To address this issue, we depart from [LLM$^+$16] by generating conjointly the parameters of the signature scheme and the ones of the commitment scheme and in particular by re-using parts of the former in the latter. More specifically, in our construction, a commitment to $\mathbf{m}$ is $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm}$, for a Gaussian randomness $\mathbf{r}$, where $\mathbf{A}$ is a matrix from the signer's public key and $\mathbf{D}$ is a public random matrix. From the efficiency standpoint, this has two important effects. First, this allows merging the randomness $\mathbf{r}$ with the other parts of the signatures, as we explain below, and thus to reduce the number of elements that we have to prove knowledge of. Second, as $\mathbf{A}$ is no longer hidden by a matrix $\mathbf{U}$, this significantly reduces the discrepancy between the adversary output and the extracted SIS solution in the security proof, leading to much better parameters.

Obviously, this has important consequences on the construction as the commitment matrix $\mathbf{A}$ is now selected by the signer, which is usually embodied by the adversary in privacy security games. To ensure that $\mathbf{A}$ is random to make the Ajtai commitment hiding, we need to generate it as a hash output. This solution is then totally incompatible with the [LLM$^+$16] approach where the signer needs to generate $\mathbf{A}$ together with an associated trapdoor.

Instead of Boyen's signature, we then choose to use the trapdoors of [MP12], which interface well with the Ajtai commitment. More precisely, our public key is composed of a random matrix $\mathbf{A}$, a matrix $\mathbf{B} = \mathbf{AR}$ and a random syndrome $\mathbf{u}$, and the secret key is a random ternary matrix $\mathbf{R}$. In order to sign a binary message $\mathbf{m}$ hidden in a commitment $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm}$, we use pre-image sampling to sample a Gaussian vector $\mathbf{v}'$ such that $[\mathbf{A}|\tau\mathbf{G} - \mathbf{B}]\mathbf{v}' = \mathbf{u} + \mathbf{c}$, where τ is a tag from a tag space $\mathcal{T} \subseteq \mathbb{Z}_q^\times$ and $\mathbf{G}$ is the gadget matrix from [MP12]. As $\mathbf{A}$ is involved in both the left hand side of the equation and in $\mathbf{c}$, we can set the signature as $(\tau, \mathbf{v} = \mathbf{v}' - [\mathbf{r}^T|\mathbf{0}]^T)$. Verification consists in checking

$$[\mathbf{A}|\tau\mathbf{G} - \mathbf{B}]\mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod q \text{ and } \|\mathbf{v}\|_\infty \text{ small.} \tag{1}$$

One can note that we have removed in the process the binary decomposition of $\mathbf{c}$. We indeed choose a very different approach in the security proof which shows that this step is actually not necessary. Removing this decomposition is also crucial in order to compact the commitment randomness $\mathbf{r}$ with the pre-image $\mathbf{v}'$. It avoids further intermediate steps that deteriorate the SIS solution extracted from the forgery, as explained above, which leads to better parameters overall. Moreover, when it comes to proving knowledge of the signature, each intermediate step makes the whole statement harder to prove and requires to create additional witnesses, i.e., each bit of $\mathbf{c}$, that must be committed, whose membership in $\{0, 1\}$ must be proven, etc. Our point here is that each seemingly innocent modification is considerably amplified when considering the full protocol and therefore results in major gains.

At this stage, a reader familiar with the construction in [dPLS18] might wonder why we do not try to embed the committed message in the tag τ, instead of having this $\mathbf{Dm}$ component in our verification equation. Here, we need to recall that the situation of [dPLS18] is very specific as the signer (the group manager in their application) knows the signed message $\mathbf{m}$, which belongs to some bounded set in their application. In our case, we want to hide this message that may have a very large entropy (this is for example the case in anonymous credentials systems). In all cases, the security reduction must guess, at the setup stage, the value of the tag τ^* involved in the forgery. Therefore, if τ is generated from $\mathbf{m}$ itself, then the reduction would have to guess this message, which would result in an exponential security loss in most scenarios. A workaround could be to construct τ from $H(\mathbf{m})$ for some appropriate function H (most likely a hash function because of the properties it would have to satisfy) whose image has lower entropy so as to guess $H(\mathbf{m})$ instead of $\mathbf{m}$. Alternatively, H could be modelled as a random oracle. The problem with this solution is that verification would now require to prove that $H(\mathbf{m})$ has been correctly evaluated. For very specific scenarios *(e.g., blind signature)* where $\mathbf{m}$ can be revealed at the verification time, this would work with a security loss depending on the entropy of $H(\mathbf{m})$. For all others (*e.g.*, group signature, anonymous credentials, e-cash, etc), where the message must remain secret, this would not be possible with the zero-knowledge frameworks we target because of the nature of H. As we aim to design a versatile tool, suitable for all applications, we choose to have a tag uncorrelated to the message, hence the $\mathbf{Dm}$ component mentioned above. As per the security proof, there are two constraints in the way to choose tags: generate tags without encountering collisions to only emit one signature per tag, and without enduring an exponential loss in the security proof due to guesses. Given that we essentially target privacy-preserving applications such as group signatures or anonymous credentials, we focus, in the body of our paper, on a stateful construction that inherently solves these two problems. For all these applications, it is indeed natural for the signer to keep track of the signatures it has issued. For group signature, this is even a requirement of the security model [BSZ05]: a registration table must be updated after each addition of a group member. However, for completeness, we show in Appendix H that our construction can easily be tweaked to be stateless, at the cost of a very mild increase of the signature size, while complying with the two constraints above.

So far, we have essentially discussed improvements of both the commitment and the signature schemes. Table 1.1 shows that our resulting signature is between 30 and 40 times smaller than that of [LLM$^+$16] when considering the same setting (standard lattices). However, this gain is still not sufficient to lead to practical proofs as ZK lattice proofs are still complex, even with the recent

	setting	λ	$\lvert\mathsf{pk}\rvert$ (MB)	$\lvert\mathsf{sk}\rvert$ (MB)	$\lvert\mathsf{sig}\rvert$ (**KB**)	$\lvert\pi\rvert$ (**KB**)
[LLM+16] (exact proof)	stand.	128	$296 \cdot 10^4$	$156 \cdot 10^2$	$862 \cdot 10^1$	$102 \cdot 10^5$
[LLM+16] (fast mode)	stand.	128	$707 \cdot 10^4$	$372 \cdot 10^2$	$139 \cdot 10^2$	$671 \cdot 10^3$
Sec. 3 (exact proof)	stand.	128	$115 \cdot 10^1$	892	261	$306 \cdot 10^3$
Sec. 3 (fast mode)	stand.	128	$296 \cdot 10^1$	$229 \cdot 10^1$	418	$177 \cdot 10^2$
Sec. 3.3 (exact proof)	module	128	7.8	8.9	273	639

Table 1.1. Comparison of efficiency estimates of the signature schemes of [LLM+16], of Section 3 and of Section 3.3 for $\lambda = 128$ bits of quantum security, with the size of zero-knowledge proof of possession of a message-signature pair. In the setting column, *stand.* stands for standard lattices, as opposed to the ring setting of our last construction. The proofs for [LLM+16] and Section 3 are either exact proofs or approximate ones using the *fast mode* of Section 4.2 and described in the technical overview. The complete analysis and parameter sets used for these estimates can be found in Appendix G.

framework of [YAZ+19]. We now focus on the proofs of knowledge necessary for our protocol and explain how we can modify the previous framework for a better efficiency.

Efficient Protocols and Zero-Knowledge Arguments. A "signature scheme with efficient protocols" requires two kinds of protocols, one to get a signature on a committed message and one for proving possession of a message-signature pair. Regarding the former, the problem is rather simple as the message $\mathbf{m}$ to sign is already embedded in a commitment $\mathbf{c} = \mathbf{A}\mathbf{r} + \mathbf{D}\mathbf{m}$. However, we have to slightly modify this construction because both the user requesting the signature and the signer must contribute to the randomness of the commitment. This leads to a commitment $\mathbf{c} = \mathbf{A}(\mathbf{r}' + \mathbf{r}'') + \mathbf{D}\mathbf{m}$ where $\mathbf{r}'$ is added by the user to enforce the hiding property of $\mathbf{c}$ and $\mathbf{r}''$ is added by the signer to be able to handle any query in the security proof. Only the former needs to prove knowledge of $\mathbf{r}'$ and $\mathbf{m}$ so as to rely on the EUF-CMA property of the signature scheme we introduced. In all cases, the user ends up with a signature $(\tau, \mathbf{v})$ on a binary $\mathbf{m}$ verifying (1) and needs to prove it in a zero-knowledge way.

For that, we employ the recent zero-knowledge framework proposed by Yang et al. [YAZ+19] which can be used to prove linear relations with quadratic constraints. The latter feature is very useful in our case as our verification equation (1) is quadratic in $(\mathbf{m}; (\tau, \mathbf{v}))$ because of the term $\tau\mathbf{G}\mathbf{v}_2$ (where $\mathbf{v}_2$ is the bottom part of $\mathbf{v}$). Moreover, this allows one to prove that an element is short by first writing its binary decomposition and then proving that each resulting component x is indeed binary through the quadratic equation $x(x-1) = 0$.

Unfortunately, this nice feature comes at a price as this decomposition procedure entails a $(\log_2 B)$-fold increase of the size of the witness $\mathbf{v}$, where B is a bound on $\|\mathbf{v}\|_\infty$. For a high dimensional vector $\mathbf{v}$ in $\mathbb{Z}^m$, this results in a very large proof which has led the authors of [YAZ+19] to propose a so-called *fast mode* for their protocol. In a nutshell, this variant relies on the observation that the norm of $\mathbf{Hv}$, for a random short matrix $\mathbf{H}$ of dimension $k \times m$, implies some bound on the norm of $\mathbf{v}$, even when the latter is chosen by the adversary. As $\mathbf{Hv}$ must be hidden, one must still use the quadratic relation above to prove shortness but on a witness with a much smaller dimension as k is in practice much smaller than m. The efficiency gains are very significant but we point out several shortcomings with the solution proposed in [YAZ+19]. First, contrarily to the claim in [YAZ+19], this fast mode *cannot* be used to prove that $\mathbf{v}$ is positive and we provide a concrete counter-example in Section 4. This is not a problem in our case as we only want to prove results on the ℓ_∞ norm of $\mathbf{v}$ but this can be a problem for specific applications such as the e-cash system considered in [YAZ+19]. Second, the authors in [YAZ+19] make use of a binary matrix $\mathbf{H}$ which significantly deteriorates the overall statement as one must set a bound $m\beta$ on the norm of $\mathbf{Hv}$, when $\|\mathbf{v}\|_\infty$ is bounded by β. Although this soundness gap seems unavoidable with this mode, we show that we can do better with a matrix $\mathbf{H} \in \{-1, 0, 1\}^{k \times m}$, which allows selecting better parameters and thus leads to more efficient protocols.

Finally, we also propose in Appendix E a series of optimizations for the protocol of [YAZ+19] that range from better parameter selection to compression of the commitments, resulting in further efficiency improvements. For a fair comparison, the figures in Table 1.1 take into account these improvements for both our scheme and the one from [LLM+16]. This table shows that our contributions reduce the size of a proof of knowledge (using the fast mode) to roughly 18 MB, which can be interpreted in two ways. On the one hand, this is a dramatic improvement over [LLM+16]. On the other hand, this is still large and probably impractical for many applications. The next part of our contributions thus investigates how to instantiate our construction in another setting to further reduce this size.

Extending to Structured Lattices. Our construction extends to the module setting where we replace the integers by polynomials with integer coefficients. More concretely, we consider a power-of-two cyclotomic ring, i.e., $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ with n a power-of-two. The additional structure yields more efficient computations, as well as more compact keys. The trapdoors of [MP12] have already been used over such algebraic rings, e.g., [DM14,dPLS18,BEP+21], which makes our module construction very similar to the one based on standard lattice assumptions. All the tools required to prove the security of our scheme also have a ring counterpart, which therefore leads to almost no differences in the security proofs either. The main difference comes when considering exact zero-knowledge proofs over algebraic rings. Our verification equation in the module setting is

$$[\mathbf{A}|\tau\mathbf{G} - \mathbf{B}]\mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod qR \text{ and } \mathbf{v} \text{ short.} \tag{2}$$

Proving knowledge of (2) requires to prove that (1) τ is in the specified tag space, (2) $\mathbf{v}$ is short, (3) $\mathbf{m}$ is a vector of binary polynomials, and (4) that the quadratic equation is verified. Based on state-of-the-art proof systems, (1) constrains which tag space to choose so that we can efficiently prove membership, while ensuring that a difference of tags is in $(R/qR)^\times$ as needed per the security proofs. Statement (2) requires to define a notion of shortness over the ring, which is usually defined based on the size of the polynomials' coefficients. Up until recently, exact proofs performing the latter task [BLS19,ENS20] (also used for (3)) used NTT packing, i.e., interpreting the coefficients of $\mathbf{v}$ as the NTT (Number Theoretic Transform) of another vector $\mathbf{v}'$, which is most efficient when $X^n + 1$ splits into low-degree irreducible factors modulo q. This splitting makes it harder to choose a proper tag space for which differences are always invertible. Finally, (4) requires a proof system able to deal with quadratic equations. Similar relations [dPLS18,LNPS21] were handled by transforming the relation quadratic in the witnesses into a linear relation in the commitment of the witnesses. Since efficient proofs of commitment opening rely on relaxed openings, this solution introduces a soundness gap in the proven statement, which we would like to avoid.

Instead, we use the very recent framework of Lyubashevsky et al. [LNP22] which provides a unified method to prove all our statements. It extends the previous works of [BLS19,ENS20] and enables proving quadratic relations exactly, as well as quadratic evaluations. The latter can be used to prove exact bounds directly in the ℓ_2 norm, which leads to more efficient proofs than proving ℓ_∞ bounds.

In the module setting, we therefore end up with a signature scheme that is efficient on all metrics, as highlighted in Table 1.1. In particular, we manage to keep our proofs of knowledge of a message-signature pair below 650 KB[6]. As these proofs are one of the main building blocks of privacy-preserving protocols, these efficiency gains readily translate to the latter and thus should have a significant impact on the area. More generally, our construction is designed to be used as a black box, which should foster many applications, as was the case with the pairing-based signatures with efficient protocols [CL04,BB08,PS16].

Application: Anonymous Credentials. Signature with efficient protocols like the one we propose gives a single signature construction that can be turned into several privacy-preserving primitives such as group signatures, anonymous credentials, e-cash etc. We give an example of one such construction based on our signature to show how it interfaces with the "efficient protocols". More precisely, we propose an anonymous credentials system following the syntax and security model from [FHS19]. At a high-level, the security relies on the zero-knowledge and soundness properties of the proof system and on the EUF-CMA security of our signature. To the best of our knowledge, this provides the first explicit lattice-based anonymous credentials system.

2 Preliminaries

Throughout this paper, for two integers $a \leq b$, we define $[a, b] = \{k \in \mathbb{Z} : a \leq k \leq b\}$. When $a = 1$ and $b \geq 1$, we simply use $[b]$ to denote $[1, b]$. For a

[6] To remain as broad as possible, we use statistical trapdoors and related tools. One could however get further efficiency gains by using a computational instantiation.

positive integer q, we define $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. In this work, we consider q to be an odd prime (or product of odd primes), and we sometimes identify $\mathbb{Z}_q$ with the set of representatives $[-(q-1)/2, (q-1)/2]$. The vectors are written in bold lowercase letters $\mathbf{a}$, while the matrices are in bold uppercase letters $\mathbf{A}$. The transpose operator is denoted with the superscript T. The identity matrix of size $n \times n$ is denoted by $\mathbf{I}_n$. For any $\mathbf{a} \in \mathbb{R}^n$, we define its Euclidean (ℓ_2) norm as $\|\mathbf{a}\|_2 = (\sum_{i\in[n]} |a_i|^2)^{1/2}$ and its infinity (ℓ_∞) norm as $\|\mathbf{a}\|_\infty = \max_{i\in[n]}|a_i|$. For a matrix $\mathbf{A} = [\mathbf{a}_i]_{i\in[m]} \in \mathbb{R}^{n\times m}$, we define $\|\mathbf{A}\|_{\max} = \max_{i\in[m]}\|\mathbf{a}_i\|_\infty$, and $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq\mathbf{0}}\|\mathbf{A}\mathbf{x}\|_2/\|\mathbf{x}\|_2$. We denote by λ the security parameter.

2.1 Lattices

A (full-rank) *lattice* $\mathcal{L}$ of rank n is a discrete additive subgroup of $\mathbb{R}^n$. The *dual lattice* of a lattice $\mathcal{L}$ is defined by $\mathcal{L}^* = \{\mathbf{x} \in \mathrm{Span}_{\mathbb{R}}(\mathcal{L}) : \forall \mathbf{y} \in \mathcal{L}, \langle \mathbf{x},\mathbf{y}\rangle \in \mathbb{Z}\}$. In this work, we consider the following family of q-ary lattices.

Definition 2.1. *Let n, m, q be positive integers. Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. We define the lattice $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q\}$.*

2.2 Probabilities

For a finite set S, we define $|S|$ to be its cardinality, and $U(S)$ to be the uniform probability distribution over S. The action of sampling $x \in S$ from a probability distribution P is denoted by $x \hookleftarrow P$. We use $x \sim P$ to say that the random variable x follows the distribution P. The *statistical distance* between two discrete probability distributions P and Q over a countable set S is defined as $\Delta(P,Q) = \frac{1}{2}\sum_{x\in S}|P(x) - Q(x)|$. We start by recalling the leftover hash lemma from [HILL99] which we write to match our context and notations.

Lemma 2.1 (Adapted from [HILL99,DORS08]). *Let n, m, q be positive integers such that q is an odd prime. For $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$, $\mathbf{x} \sim U(\{-1,0,1\}^m)$, and $\mathbf{u} \sim U(\mathbb{Z}_q^n)$, it holds that $\Delta((\mathbf{A}, \mathbf{A}\mathbf{x} \bmod q), (\mathbf{A}, \mathbf{u})) \leq \frac{1}{2}\sqrt{q^n/3^m}$. In particular, whenever $m \log_2 3 \geq n \log_2 q + \omega(\log_2 \lambda)$, the statistical distance is negligible.*

For any *center* vector $\mathbf{c} \in \mathbb{R}^n$, and *Gaussian width* $\sigma > 0$, we define the Gaussian function $\rho_{\sigma,\mathbf{c}} : \mathbf{x} \in \mathbb{R}^n \mapsto \exp(-\pi\|\mathbf{x} - \mathbf{c}\|_2^2/\sigma^2)$. For a lattice $\mathcal{L}$ of rank n, we define the *discrete Gaussian distribution* $\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}}$ of support $\mathcal{L}$, width σ and center $\mathbf{c}$ by $\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}} : \mathbf{x} \in \mathcal{L} \mapsto \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(\mathcal{L})$, where $\rho_{\sigma,\mathbf{c}}(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. When $\mathbf{c} = \mathbf{0}$, we omit it in the notations. We then use it to define the *smoothing parameter* of a lattice $\mathcal{L}$ [MR07], parameterized by a real $\varepsilon > 0$, by $\eta_\varepsilon(\mathcal{L}) = \inf\{s > 0 : \rho_{1/s}(\mathcal{L}^*) \leq 1+\varepsilon\}$. If the standard deviation is wider than the smoothing parameter, the discrete Gaussian distribution benefits from properties that are similar to the ones of the continuous Gaussian distribution. In particular, the sum of two independent discrete Gaussians is a discrete Gaussian.

Lemma 2.2 (Adapted from [Reg05, Claim 3.9][MP13, Thm. 3.3]). *Let $\mathcal{L}$ be lattice of rank n. Let $r, s > 0$ and $t = \sqrt{r^2 + s^2}$ be such that $rs/t \geq \eta_\varepsilon(\mathcal{L})$ for* *some $\varepsilon \in (0, 1/2]$. Then, we have $\Delta(\mathcal{D}_{\mathcal{L},r} + \mathcal{D}_{\mathcal{L},s}, \mathcal{D}_{\mathcal{L},t}) \leq 7\varepsilon/4$. The condition on r, s is satisfied for example when $r, s \geq \sqrt{2}\eta_\varepsilon(\mathcal{L})$.*

When centered around $\mathbf{0}$, the discrete Gaussian distribution benefits from tail bounds similar to the standard Gaussian distribution. In this work, we use tail bounds on the ℓ_2 and ℓ_∞ norms. We also recall the result of [Lyu12] bounding the magnitude of $\langle \mathbf{x}, \mathbf{v} \rangle$ for a discrete Gaussian $\mathbf{x}$ and an arbitrary vector $\mathbf{v}$. Although the tail bound on the ℓ_∞ norm follows directly from the latter, it was first proven in [Pei08, Cor. 5.3].

Lemma 2.3 ([Ban93, Lem. 1.5][Pei08, Cor. 5.3][Lyu12, Lem 4.3]). *Let $\mathcal{L}$ be a lattice of rank n. Let $\sigma > 0$ and $\mathbf{v} \in \mathbb{R}^n$. Then, for all $t > 0$, it holds that*

1. $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L},\sigma}}\left[\|\mathbf{x}\|_2 > \sigma\sqrt{n}\right] < 2^{-2n}$,
2. $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L},\sigma}}\left[\|\mathbf{x}\|_\infty > \sigma \log_2 n\right] \leq 2ne^{-\pi \log_2^2 n}$,
3. $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L},\sigma}}\left[|\langle \mathbf{x}, \mathbf{v} \rangle| > \sigma t \|\mathbf{v}\|_2\right] \leq 2e^{-\pi t^2}$.

We also use the following bound on the spectral norm of a matrix with independent sub-Gaussian entries. We recall the definition of a sub-Gaussian random vector.

Definition 2.2 (Sub-Gaussian Distribution). *Let n be a positive integer, and $\mathbf{x}$ a (discrete or continuous) random vector over $\mathbb{R}^n$. We say that $\mathbf{x}$ is sub-Gaussian with sub-Gaussian moment s if for all unit vector $\mathbf{u} \in \mathbb{R}^n$ and all $t \in \mathbb{R}$, we have $\mathbb{E}[\exp(t\langle \mathbf{x}, \mathbf{u} \rangle)] \leq e^{s^2t^2/2}$.*

Lemma 2.4 ([Ver12]). *Let ℓ, m be two positive integers, and $\mathcal{P}$ a sub-Gaussian distribution of moment s. There exists a universal constant $C > 0$ such that for all $t > 0$, $\mathbb{P}_{\mathbf{U} \hookleftarrow \mathcal{P}^{\ell \times m}}[\|\mathbf{U}\|_2 \geq Cs(\sqrt{\ell} + \sqrt{m} + t)] \leq 2e^{-\pi t^2}$.*

By noticing that $\mathcal{P} = U([-1, 1])$ is sub-Gaussian with moment $\sqrt{2/3}$, we can bound the spectral norm of ternary uniform matrix by $C\sqrt{2/3}(\sqrt{\ell} + \sqrt{m} + t)$ except with probability $2e^{-\pi t^2}$, for some constant $C > 0$ that does not depend on the dimensions. We can verify experimentally that in this case $C\sqrt{2/3} \leq 1$, and we thus omit it in the rest of the paper for clarity. The security proof of our signature requires a bound on $\|\mathbf{Um}\|_2$ for an arbitrary message $\mathbf{m} \in \{0, 1\}^m$ and uniform ternary $\mathbf{U}$. When m is small, Lemma 2.4 gives a close to optimal bound by $\|\mathbf{Um}\|_2 \leq \|\mathbf{U}\|_2\sqrt{m}$. However, when m is large, we expect a tighter bound. By using the fact that square sub-Gaussian random variables are sub-exponential and tail bounds on sub-exponential distributions, we get the following lemma. The proof and associated definitions are provided in Appendix A.

Lemma 2.5. *Let ℓ, m be two positive integers and $x > 0$. We assume that $\ell > x \cdot 10/\log_2 e$. Let $\mathbf{m} \in \{0, 1\}^m$. We have $\mathbb{P}_{\mathbf{U} \hookleftarrow U([-1,1])^{\ell \times m}}[\|\mathbf{Um}\|_2 \geq 2\sqrt{\ell m}] \leq 2^{-x}$.*

In our situation, $x = \Theta(\lambda)$ with λ the security parameter, and $\ell = O(n \log_2 q + \omega(\log_2 \lambda))$. The condition $\ell > 10x/\log_2 e$ is then verified. Note that this condition is necessary only to obtain the simple bound $2\sqrt{\ell m}$ with probability 2^{-x}, but one could use a different bound or different probability to avoid this condition. Combining both lemmas gives the following

$$\mathbb{P}_{\mathbf{U} \hookleftarrow U([-1,1]^{\ell \times m})}[\|\mathbf{U}\mathbf{m}\|_2 \geq \min(2\sqrt{\ell}, \sqrt{\ell} + \sqrt{m} + t)\sqrt{m}] \leq 2^{-2\lambda} + 2e^{-\pi t^2}, \quad (3)$$

whenever $\ell \geq 20\lambda / \log_2 e$ which is the case in our context. The spectral bound of Lemma 2.4 is also necessary to set the correct parameters to sample Gaussian vectors $\mathbf{v}$ verifying $[\mathbf{A}|\tau\mathbf{G} - \mathbf{A}\mathbf{R}]\mathbf{v} = \mathbf{u}$, where $\mathbf{A}$ is a uniform matrix, $\mathbf{R}$ a short random matrix and $\mathbf{G}$ the gadget matrix of [MP12] used for efficient pre-image sampling. Our signature uses the following pre-image sampling algorithm.

Lemma 2.6 ([MP12]). *There exists an algorithm* SampleD *that takes as input a trapdoor matrix* $\mathbf{R} \in \mathbb{Z}^{m_1 \times n\lceil \log_2 q \rceil}$*, a partial parity-check matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m_1}$*, an invertible tag matrix* $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$*, a syndrome* $\mathbf{u} \in \mathbb{Z}_q^n$ *and a standard deviation* $\sigma \geq \eta_\varepsilon(\mathbb{Z})\sqrt{7}\sqrt{1 + \|\mathbf{R}\|_2^2}$*, and that outputs* $\mathbf{v}$ *that is statistically close to* $\mathcal{D}_{\mathbb{Z}^{m_1 + n\lceil \log_2 q \rceil}, \sigma}$ *conditioned on* $[\mathbf{A}|\mathbf{H}\mathbf{G} - \mathbf{A}\mathbf{R}]\mathbf{v} = \mathbf{u} \bmod q$*, with* $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}$ *and* $\mathbf{g} = [1 \ldots 2^{\lceil \log_2 q \rceil - 1}]$*.*

2.3 Hardness Assumption

The security of our signature scheme relies on the *Short Integer Solution* (SIS) problem [Ajt96], which we recall here.

Definition 2.3 (Short Integer Solution). *Let* n, m, q *be positive integers, and* $\beta_2 \geq \beta_\infty \geq 1$*. The* Short Integer Solution *problem* $\mathrm{SIS}^{\infty,2}_{n,m,q,\beta_\infty,\beta_2}$ *asks to find* $\mathbf{x} \in \mathcal{L}_q^\perp(\mathbf{A}) \backslash \{\mathbf{0}\}$ *given* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ *such that* $\|\mathbf{x}\|_\infty \leq \beta_\infty$ *and* $\|\mathbf{x}\|_2 \leq \beta_2$*.*

Note that the original formulation of SIS considers a single bound β on the ℓ_2 norm. There is a trivial reduction from the latter to $\mathrm{SIS}^{\infty,2}_{n,m,q,\beta_\infty,\beta_2}$ by setting $\beta = \min(\beta_\infty\sqrt{m}, \beta_2)$. As discussed by Micciancio and Peikert [MP13, Thm. 1.1], using both norm bounds leads to more precise hardness results, and sometimes smaller approximation factors when relating the problem to worst-case problems on lattices. Moreover, it seems to be relevant for the concrete hardness of the problem as well. Indeed, most lattice reduction algorithms aim at finding vectors in the ball of radius β_2 but without constraining the magnitude of the coefficients. Finding a lattice vector in the intersection of the ball of radius β_2 and the hypercube of half side β_∞ is at least as hard as the same task without the β_∞ bound. When $\beta_\infty \ll \beta_2$, it may even be substantially harder.

2.4 Signature Scheme

A signature scheme is defined by four algorithms. The Setup algorithm is a probabilistic algorithm that, on input a security parameter λ, outputs the public parameters pp that will be common to all users. The key generation algorithm KeyGen is a probabilistic algorithm that, on input pp, outputs a secret signing key sk and a public verification key pk. The signing algorithm Sign is a probabilistic algorithm which, on inputs sk and a message $\mathbf{m}$ (and pk, pp), outputs a signature sig. Finally, the verification algorithm Verify is a deterministic algorithm that, on inputs $\mathsf{pk}, \mathbf{m}, \mathsf{sig}$ (and pp), outputs 1 if sig is a valid signature on $\mathbf{m}$ under pk, and 0 otherwise. We use the *Existential Unforgeability against Chosen Message Attacks* (EUF-CMA) security model, which we formally recall in Appendix B along with the security proofs of our signature scheme.

3 A Lattice-Based Signature Scheme

We present here our signature scheme which interfaces smoothly with privacy-enhancing protocols. It provides an alternative to the only such scheme based on lattices due to Libert et al. [LLM$^+$16].

One of the main differences between their construction and ours is that we aim at optimizing the interactions between the commitment scheme implicitly used by such kind of protocols and the signature scheme itself. In [LLM$^+$16], the public parameters of these two components were generated independently. We depart completely from this approach by generating these parameters conjointly and even by using a common matrix $\mathbf{A}$ for these two parts. Besides the natural gain in the public key size, this strategy allows one to merge different components of the signature itself. In particular, compared to [LLM$^+$16], our signature no longer has to include the commitment opening, which significantly reduces its size.

Obviously, this has important consequences on the design of the scheme itself. One of them is that it forbids to re-use the approach of [LLM$^+$16], inherited from Boyen signature [Boy10], where $\mathbf{A}$ was generated together with a trapdoor, because it would clearly break the hiding property of the commitment scheme. We instead rely on a $\mathbf{G}$-trapdoor $\mathbf{R}$ of size $m_1 \times m_2$ in the sense of [MP12] and then use a matrix $[\mathbf{A}|\tau\mathbf{G} - \mathbf{AR}]$ where τ is a tag from $\mathbb{Z}_q^\times$. We can therefore generate $\mathbf{A}$ as a random matrix[7] of size $n \times m_1$, where m_1 is the dimension of the commitment randomness. We then use it to construct the commitment $\mathbf{c}$ to a message $\mathbf{m} \in \{0,1\}^{m_3}$ as $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm} \bmod q$, where $\mathbf{D}$ is a random matrix of size $n \times m_3$ and m_3 is the dimension of the message. The randomness $\mathbf{r}$ can then be merged with the short vector $\mathbf{v}$ generated thanks to the trapdoor, as mentioned above.

In [LLM$^+$16], the authors had to first compute a binary decomposition $\mathbf{c}'$ of the commitment $\mathbf{c}$ to the message before generating a short pre-image of $\mathbf{u} + \mathbf{Dc}'$ where $\mathbf{u}$ (resp. $\mathbf{D}$) was some public vector (resp. matrix). This might look harmless when we only consider the signature because it does not increase its size. However, when plugged in the Yang et al. ZK framework [YAZ$^+$19] this replaces one secret vector $\mathbf{c}$ by $\log_2 q$ ones and makes the overall statement to prove more complex[8]. To remove this binary decomposition we revisit the

[7] In our protocol for signing hidden messages, we will have to enforce this requirement but this can be done easily by setting $\mathbf{A}$ as some hash output.

[8] Considering a binary tag also leads to similar inefficiencies

security proof and show how to avoid it by using an argument based on the Rényi Divergence. Additionally, this change seems necessary to extend our construction to polynomial rings, as described in Section 3.3.

More generally, all the modifications we introduce have a second positive effect on complexity. In both our security proof and the one of [LLM+16], it is necessary to generate the public matrices with hidden relations, usually by multiplying one by some low-norm matrix $\mathbf{U}$ to generate the other. This impacts the norm of the extracted solutions, which grows with the number of such matrices and computational steps, and therefore impacts the system parameters. By reusing $\mathbf{A}$ for different purposes and by removing some computational steps (e.g., multiplication by $\mathbf{D}$), we significantly reduce the discrepancy between the adversary output and the resulting SIS solution, leading to much better parameters.

3.1 Description of the Signature

We now describe the four algorithms that define our signature scheme. The signature is designed to sign a binary message $\mathbf{m}$. We present our scheme for the more general case of a message with variable length rather than a variable number of blocks of fixed length which may require unnecessary padding.

```
Algorithm 3.1: Setup
Input: Security parameter $\lambda$.
  Choose a positive integer $n$.
  Choose a prime integer $q$.
  Choose a positive integer $q' \leq q$.                                    ▷ Bound on tags
  $\mathcal{T} \leftarrow \mathbb{Z}_{q'} \setminus \{0\}$.                                                ▷ Tag space
  Choose $f(\lambda) \leftarrow \omega(\log_2 \lambda)$.
  $m_1 \leftarrow \lceil (n \log_2 q + f(\lambda)) / \log_2 3 \rceil$.              ▷ Commitment randomness dimension
  $m_2 \leftarrow n \lceil \log_2 q \rceil$.
  $m \leftarrow m_1 + m_2$.                                              ▷ Signature dimension
  Choose a positive integer $m_3$.                                ▷ Maximum bit-size of $\mathbf{m}$
  $\mathbf{g} \leftarrow [2^0 | \dots | 2^{\lceil \log_2 q \rceil - 1}] \in \mathbb{Z}_q^{1 \times \lceil \log_2 q \rceil}$.                          ▷ Gadget vector
  $r \leftarrow \eta_\varepsilon(\mathbb{Z})$.                                     ▷ $r = 5.4$ leads to $\varepsilon \approx 2^{-131}$
  Choose $t > 0$.
  $\sigma \leftarrow r\sqrt{7}\sqrt{(\sqrt{m_1} + \sqrt{m_2} + t)^2 + 1}$.                 ▷ Preimage sampling width
  $\sigma_2 \leftarrow \max\left(\sqrt{m_3 \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_2} + t)^2 - \sigma^2}, \omega(\sqrt{\log_2 m_1})\right)$.
  $\sigma_1 \leftarrow \sqrt{\sigma^2 + \sigma_2^2}$.
  $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m_3})$.                             ▷ Message commitment key
Output: $\mathsf{pp} = (\mathbf{D}; \mathbf{g}; \lambda, n, m_1, m_2, m_3, q, \sigma, \sigma_2, \sigma_1)$.
```

```
Algorithm 3.2: KeyGen
Input: Public parameters pp as in Algorithm 3.1.
  $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m_1})$.
  $\mathbf{R} \hookleftarrow U([-1, 1]^{m_1 \times m_2})$.
  $\mathbf{B} \leftarrow \mathbf{AR} \bmod q \in \mathbb{Z}_q^{n \times m_2}$.
  $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.
Output: $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, and $\mathsf{sk} = \mathbf{R}$.
```

Algorithm 3.3: Sign

```
Input: Signing key sk, Message $\mathbf{m} \in \{0,1\}^{m_3}$, Public key pk, Public Param. pp, State st.
  $\mathbf{r} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{m_1},\sigma_2}$.
  $\mathbf{c} \leftarrow \mathbf{Ar} + \mathbf{Dm} \bmod q$.                              ▷ Commitment to $\mathbf{m}$
  $\tau \leftarrow F(\mathsf{st})$.                                                   ▷ $\tau \in \mathcal{T}$
  $\mathbf{v} \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \tau\mathbf{I}_n, \mathbf{u}+\mathbf{c}, \sigma) - [\mathbf{r}^T|\mathbf{0}_{m_2}]^T$.   ▷ $\mathbf{A}_\tau = [\mathbf{A}|\tau(\mathbf{I}_n \otimes \mathbf{g}) - \mathbf{B}]$
  $\mathsf{st} \leftarrow \mathsf{st} + 1$.
Output: $\mathsf{sig} = (\tau, \mathbf{v})$.
```

Algorithm 3.4: Verify

```
Input: Public key pk, Message $\mathbf{m} \in \{0,1\}^{m_3}$, Signature sig, Public Param. pp.
  $\mathbf{A}_\tau \leftarrow [\mathbf{A}|\tau(\mathbf{I}_n \otimes \mathbf{g}) - \mathbf{B}] \in \mathbb{Z}_q^{n \times m}$.
  Split $\mathbf{v}$ into $[\mathbf{v}_1^T|\mathbf{v}_2^T]^T$, with $\mathbf{v}_1 \in \mathbb{Z}^{m_1}$, $\mathbf{v}_2 \in \mathbb{Z}^{m_2}$.
  $b \leftarrow (\mathbf{A}_\tau \mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod q) \wedge (\|\mathbf{v}_1\|_\infty \leq \sigma_1 \log_2 m_1) \wedge (\|\mathbf{v}_2\|_\infty \leq \sigma \log_2 m_2) \wedge (\tau \in \mathcal{T})$
Output: $b$.                                        ▷ $b = 1$ if valid, 0 otherwise
```

The correctness of the signature scheme simply relies on the sum of discrete Gaussians (Lemma 2.2) and the Gaussian tail bound (Lemma 2.3). The former guarantees that $\mathbf{v}_1$ is statistically close to $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1}$, and the latter ensures that for an honest signature it holds that $\|\mathbf{v}_1\|_\infty \leq \sigma_1 \log_2 m_1$, and $\|\mathbf{v}_2\|_\infty \leq \sigma \log_2 m_2$ with overwhelming probability. Note that the randomness $\mathbf{r}$ used to commit to the message can be drawn from a Gaussian with any width $\sigma_2 > 0$. However, the security proofs require σ_1 to be at least $\min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3}$ in order to hide the shifted center of the Gaussian vector, which in turns restricts the value of σ_2. Additionally, the goal of this signature scheme being to allow signing on committed messages, σ_2 must be chosen so that the commitment scheme is statistically hiding, which is why we take it at least $\omega(\sqrt{\log_2 m_1})$. We present our signature scheme in the most general way, thus explaining the multitude of dimensions m_i and Gaussian widths σ_i. This also allows fine-tuning of the parameters depending on the specific application. Typically, an application requiring to sign only small messages of constant bit-size m_3 would be able to select a much smaller σ_1 and would then yield smaller signatures.

We also point out the fact that we express the shortness condition on $\mathbf{v}$ in the ℓ_∞ norm. This is due to the fact that the zero-knowledge argument framework from [YAZ$^+$19] that we consider to prove possession of a message-signature pair allows one to prove bounds on the coefficients more naturally. As a result, we can base the security of our signature scheme on $\text{SIS}^{\infty,2}$ which is at least as hard as SIS^2 as explained in Section 2.3.

An example parameter set, also taking into account the requirements of Sections 4 and 5, can be found in Appendix G, Table G.2. The scheme can also be instantiated as a standalone signature, without considering the efficient protocols and zero-knowledge proof systems. This would allow one to reduce the size of q, but at the expense of increasing n to achieve the same security, which in the end leads to similar signature and key sizes.

Remark 3.1. As discussed in Section 1, we choose to describe a stateful version of our construction that better suits our applications, hence the fact that our tags τ are generated as $F(\mathsf{st})$. The only requirements placed on F are that it must be injective, with outputs in the tag space, which should easily be met in practice. For example, in the case of group signatures, one can proceed as in [dPLS18] and set the tags as the group members' identities. Nevertheless, if selecting such a function F proved to be difficult for some use case, we recall that a stateless version of our construction is provided in Appendix H.

3.2 Security of the Signature

We distinguish two types of forgeries that an attacker can produce, which we treat separately for the sake of clarity. More precisely we distinguish between the cases depending on whether or not the tag τ^* of the forgery has been re-used from the signature queries. Combining the corresponding lemmas proves the EUF-CMA security of the signature under the SIS assumption. It consists in the SIS challenger tossing a coin and proceeding as in either Lemma 3.1 or 3.2 and aborting if the forgery does not match the coin toss. The proofs are provided in Appendix B.2 and B.3 for completeness.

Lemma 3.1. *An adversary produces a* Type I *forgery* $(\tau^*, \mathbf{v}^*)$ *if the tag* τ^* *does not collide with the tags of the signing queries. If an adversary can produce a Type I forgery with advantage* δ*, then we can construct an adversary* $\mathcal{B}$ *that solves the* $\mathrm{SIS}^{\infty,2}_{n,m_1+1,q,\beta_\infty,\beta_2}$ *problem with advantage* $Adv[\mathcal{B}] \gtrsim \delta/(|\mathcal{T}| - Q)$*, for*

$$\begin{cases} \beta_\infty &= \sigma_1 \log_2 m_1 + m_2 \sigma \log_2 m_2 + m_3 \\ \beta_2 &= \sqrt{1 + (\sqrt{m_1} + \sqrt{m_2} + t)^2} \cdot \sqrt{m_1(\sigma_1 \log_2 m_1)^2 + m_2(\sigma \log_2 m_2)^2} \\ &\quad + \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3} + 1. \end{cases}$$

Lemma 3.2. *An adversary produces a* Type II *forgery* $(\tau^*, \mathbf{v}^*)$ *if the tag* τ^* *is re-used from some* i^**-th signing query* $(\tau^{(i^*)}, \mathbf{v}^{(i^*)})$*, i.e.,* $\tau^* = \tau^{(i^*)}$*. If an adversary can produce a Type II forgery with advantage* δ*, we can construct* $\mathcal{B}$ *solving* $\mathrm{SIS}^{\infty,2}_{n,m_1,q,\beta'_\infty,\beta'_2}$ *with advantage*

$$Adv[\mathcal{B}] \gtrsim \frac{\delta^{\alpha^*/(\alpha^*-1)} e^{-\alpha^* \pi}}{Q},$$

for

$$\begin{cases} \beta'_\infty &= 2\sigma_1 \log_2 m_1 + m_2 \cdot 2\sigma \log_2 m_2 + m_3 \\ \beta'_2 &= \sqrt{1 + (\sqrt{m_1} + \sqrt{m_2} + t)^2} \cdot \sqrt{\sigma_1^2 m_1(1 + \log_2^2 m_1) + \sigma^2 m_2(1 + \log_2^2 m_2)} \\ &\quad + \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3}. \end{cases}$$

and where $\alpha^* = 1 + \sqrt{\log_2(1/\delta)/(\pi \log_2 e)}$*.*

3.3 Our Signature on Modules

The results of Table 1.1 show that the performances of the signature scheme from Section 3.1 and associated protocols are dramatically improved over [LLM$^+$16]. However, the complexity is still rather high and we therefore investigate in this section a way to decrease it. Concretely, we show that the signature scheme from Section 3.1 can be extended over the ring of integers of a number field. For the zero-knowledge arguments required by the efficient protocols, we employ the recent framework from [LNP22], which we detail in Section 4.3. We use a tag space that corresponds to the identity space of their group signature construction. We also use a message space that is similar to the latter but with no restriction on the number of non-zero coefficients. We present our construction with a single power-of-two cyclotomic ring, but we note that it can be adapted to use subrings for efficiency gains. For more details on the use of subrings, we refer to [LNPS21,LNP22]. In what follows, we take n a power of two and R the $2n$-th cyclotomic ring, i.e., $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$. We also define $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ for any modulus $q \geq 2$. We call θ the coefficient embedding of R, i.e., for all $r = \sum_{i\in[0,n-1]} r_i X^i \in R$, $\theta(r) = [r_0 \dots r_{n-1}]^T$. We then define $S_{bin} = \theta^{-1}(\{0,1\}^n)$ and $S_1 = \theta^{-1}(\{-1,0,1\}^n)$. We also define the usual norms $\|\cdot\|_p$ over R by $\|r\|_p := \|\theta(r)\|_p$. Finally, we define the discrete Gaussian distribution over R by $\theta^{-1}(\mathcal{D}_{\theta(R),\sigma})$, which we denote by $\mathcal{D}_{R,\sigma}$.

Remark 3.2. The Gaussian distributions are defined with respect to the coefficient embedding θ. Theoretical works usually define Gaussian distributions with respect to the Minkowski embedding (or canonical embedding) σ_H. We refer to [LPR13] for more details. In our specific case of power-of-two cyclotomic rings, it holds that $\sigma_H = \sqrt{n}\mathbf{P}\theta$ where $\mathbf{P}$ is a unitary matrix. Hence, by denoting $\mathcal{D}^{\theta}_{R,\sigma}$ (resp. $\mathcal{D}^{\sigma_H}_{R,\sigma}$) the Gaussian distribution with respect to θ (resp. σ_H), we can show that $\mathcal{D}^{\sigma_H}_{R,\sigma\sqrt{n}}$ is exactly the same distribution as $\mathcal{D}^{\theta}_{R,\sigma}$.

3.3.1 Description. Our module signature scheme is described by Algorithms 3.5, 3.6, 3.7 and 3.8.

Algorithm 3.5: Setup

```
Input: Security parameter $\lambda$.
Choose a positive integer $d$.
Choose $k \leq n$ to be a power of two.
Choose a prime integer $q$ such that $q = 2k + 1 \bmod 4k$ and $q \geq (2\sqrt{k})^k$.
Choose positive integers $w, \kappa$.
$\mathcal{T}_w \leftarrow \{\tau \in S_{bin} : \|\tau\|_2 = \sqrt{w}\}$.          ▷ Tag space
$g \leftarrow \lceil q^{1/\kappa} \rfloor$.
$m_1 \leftarrow \lceil (d \log_2 q + f(\lambda))/\log_2 3 \rceil$          ▷ $f(\lambda) = \omega(\log_2 \lambda)$
$m_2 \leftarrow d\kappa$
$m \leftarrow m_1 + m_2$.          ▷ Signature dimension
Choose a positive integer $m_3$.          ▷ Maximum bit-size of $\mathbf{m}$ is $n \cdot m_3$
$\mathbf{g} = [1 \cdots g^{\kappa-1}] \in R_q^{1\times\kappa}$.          ▷ Gadget vector
$r \leftarrow \eta_\varepsilon(\mathbb{Z})$.          ▷ $r = 5.4$ leads to $\varepsilon \approx 2^{-131}$
Choose $t > 0$.          ▷ Spectral norm slack
```

```
$\sigma \leftarrow r\sqrt{g^2+1}\sqrt{(\sqrt{nm_1}+\sqrt{nm_2}+t)^2+1}$.        ▷ Pre-image sampling width
$\sigma_2 \leftarrow \sqrt{(\sqrt{nm_1}+\sqrt{nm_3}+t)^2 \cdot nm_3 - \sigma^2}$.        ▷ Commitment randomness width
$\sigma_1 \leftarrow \sqrt{\sigma^2+\sigma_2{}^2}$.
$\mathbf{D} \hookleftarrow U(R_q^{d\times m_3})$.        ▷ Message Commitment Key
Output: $\mathsf{pp} = (\mathbf{D}; \mathbf{g}; \lambda, n, d, m_1, m_2, m_3, q, w, \kappa, \sigma, \sigma_2, \sigma_1)$.
```

Algorithm 3.6: KeyGen

```
Input: Public parameters $\mathsf{pp}$ as in Algorithm 3.5.
$\mathbf{A} \hookleftarrow U(R_q^{d\times m_1})$.
$\mathbf{R} \hookleftarrow U(S_1^{m_1\times m_2})$.
$\mathbf{B} \leftarrow \mathbf{AR} \bmod qR \in R_q^{d\times m_2}$.
$\mathbf{u} \hookleftarrow U(R_q^d)$.
Output: $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, and $\mathsf{sk} = \mathbf{R}$.
```

Algorithm 3.7: Sign

```
Input: Signing key $\mathsf{sk}$, Message $\mathbf{m} \in S_{bin}^{m_3}$, Public key $\mathsf{pk}$, Public Parameters $\mathsf{pp}$, State $\mathsf{st}$
$\mathbf{r} \hookleftarrow \mathcal{D}_{R^{m_1},\sigma_2}$.
$\mathbf{c} \leftarrow \mathbf{Ar} + \mathbf{Dm} \bmod qR$.        ▷ Commitment to $\mathbf{m}$
$\tau \leftarrow F(\mathsf{st})$.        ▷ $\tau \in \mathcal{T}_w$
$\mathbf{v} \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \tau\mathbf{I}_d, \mathbf{u}+\mathbf{c}, \sigma) - [\mathbf{r}^T|\mathbf{0}_{m_2}]^T$.        ▷ $\mathbf{A}_\tau = [\mathbf{A}|\tau(\mathbf{I}_d \otimes \mathbf{g}) - \mathbf{B}]$
$\mathsf{st} \leftarrow \mathsf{st} + 1$.
Output: $\mathsf{sig} = (\tau, \mathbf{v})$.
```

Algorithm 3.8: Verify

```
Input: Public key $\mathsf{pk}$, Message $\mathbf{m} \in S_{bin}^{m_3}$, Signature $\mathsf{sig}$, Public Parameters $\mathsf{pp}$.
$\mathbf{A}_\tau \leftarrow [\mathbf{A}|\tau(\mathbf{I}_d \otimes \mathbf{g}) - \mathbf{B}] \in R_q^{d\times m}$.
$b \leftarrow (\mathbf{A}_\tau\mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod qR) \wedge (\|\mathbf{v}\|_2 \leq \sqrt{\sigma_1^2 nm_1 + \sigma^2 nm_2}) \wedge (\tau \in \mathcal{T}_w)$
Output: $b$.        ▷ $b = 1$ if valid, $0$ otherwise
```

3.3.2 Security Analysis. The security of the scheme is now based on the problem M-SIS$_{d,m_1,q,\beta}$. It asks to find $\mathbf{w} \in R^{m_1}$ such that $\mathbf{Aw} = \mathbf{0} \bmod qR$ and $0 < \|\mathbf{w}\|_2 \leq \beta$ given $\mathbf{A} \hookleftarrow U(R_q^{d\times m_1})$. The security proofs rigorously follow that of Lemma 3.1 and 3.2. This is due to the fact that all the tools that we use have a ring counterpart. We briefly explain what tools are needed to carry out the proofs in the module case. We stress that the construction can also be used over rings ($d = 1$).

First, we need to ensure that a difference of distinct tags is invertible in R_q. By [LS18, Cor. 1.2], when $q = 2k+1 \bmod 4k$, a ring element r is invertible in R_q if $0 < \|r\|_\infty \leq q^{1/k}/\sqrt{k}$. We chose q so that a difference of tags $\tau_1 - \tau_2$ has ℓ_∞ norm at most $2 \leq q^{1/k}/\sqrt{k}$. Hence, a difference of distinct tags is in $R_q^\times$. Then, the leftover hash lemma of Lemma 2.1 has been adapted to general rings of integer by Boudgoust et al. and further generalized in [BJRW23]. We state it here for our specific usage in power-of-two cyclotomic rings.

Lemma 3.3 ([BJRW23, Lem. 2.8]). *Let $R = \mathbb{Z}[X]/\langle X^n+1\rangle$ with n a power of two, and d, m, q be positive integers with q prime. Then, $\Delta((\mathbf{A}, \mathbf{As}), (\mathbf{A}, \mathbf{u})) \leq \frac{1}{2}\sqrt{(1+q^d/3^m)^n - 1}$, where $\mathbf{A} \sim U(R_q^{d\times m})$, $\mathbf{s} \sim U(S_1^d)$ and $\mathbf{u} \sim U(R_q^d)$.*

The use of the Rényi divergence in the proof of Lemma 3.2 applies on the discrete Gaussian distributions, which are defined by their embedding to $\mathbb{R}^n$. As such, the argument remains unchanged. We also need to argue that for $\mathbf{A} \hookleftarrow U(R_q^{d\times m_1+m_2})$ and $\mathbf{v} \hookleftarrow \mathcal{D}_{R^{m_1+m_2},\mathbf{\Sigma}}$ with $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 \mathbf{I}_{nm_1} & \mathbf{0} \\ \mathbf{0} & \sigma \mathbf{I}_{nm_2} \end{bmatrix}$, then $\mathbf{u} = \mathbf{Av} \bmod q$ is close to uniform. For that, we use [LPR13, Thm. 7.4] which states that if $\sigma, \sigma_1 \geq 2nq^{(d+2/n)/(m_1+m_2)}$, then the public syndrome $\mathbf{u}$ is close to uniform in R_q^d. We note that this results holds when the Gaussian over R is defined with respect to the Minkowski embedding. As explained in Remark 3.2, in the case of our Gaussian distributions, we only need $\sigma, \sigma_1 \geq 2\sqrt{n}q^{\frac{d+2/n}{m_1+m_2}}$. Since $m_1 + m_2 \geq d(\log_2(q)/\log_2(3) + \kappa) + f(\lambda)/\log_2(3)$, the result holds whenever $\sigma, \sigma_1 \geq 3^{1+2/n} \cdot 2\sqrt{n}$, which is the case in our context.

Finally, we need to bound the spectral norm of structured matrices that are of size $nm_1 \times nm_2$ (or $nm_1 \times nm_3$). In power-of-two cyclotomic rings, the structured matrix considered is a block matrix whose blocks are nega-circulant matrices of size $n \times n$. The entries are thus all distributed according to $U([-1, 1])$ but they are not all independent within a block. This means we cannot apply Lemma 2.4 directly. The spectral norm of such a structured matrix of size $nm_1 \times nm_2$ is proven to be the maximal spectral norm of the n complex-embedded matrices of size $m_1 \times m_2$ [BJRW23, Lem. 2.3], which all have i.i.d. entries that are sub-Gaussian of moment $\sqrt{2n/3}$. Applying Lemma 2.4 to these embedded matrices with the union bound (on half the complex embeddings) gives

$$\mathbb{P}_{\mathbf{R}\hookleftarrow S_1^{m_1\times m_2}}[\|\mathbf{R}\|_2 \geq C\sqrt{2n/3}(\sqrt{m_1} + \sqrt{m_2} + t)] \leq ne^{-\pi t^2},$$

for an absolute constant $C > 0$. Although this bound is proven, we can verify experimentally that it is not tight, and rather that the original bound (when there is no structure) of $\sqrt{nm_1} + \sqrt{nm_2} + t$ for a small t (typically $6 - 7$) is satisfied with overwhelming probability. Further, we use the latter bound for setting parameters in the description of the signature.

Lemma 3.4. *If an adversary can produce a Type I forgery with advantage δ, then we can construct $\mathcal{B}$ that solves* M-SIS$^2_{d,m_1+1,q,\beta}$ *with advantage* $Adv[\mathcal{B}] \gtrsim \delta/(|\mathcal{T}_w| - Q)$*, for*

$$\begin{aligned}\beta = &\sqrt{1 + (\sqrt{nm_1} + \sqrt{nm_2} + t)^2}\sqrt{\sigma_1^2 nm_1 + \sigma^2 nm_2} \\ &+ (\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3} + 1.\end{aligned}$$

Lemma 3.5. *If an adversary can produce a Type II forgery with advantage δ, we can construct $\mathcal{B}$ that solves the* M-SIS$^2_{d,m_1,q,\beta'}$ *problem with advantage*

$$Adv[\mathcal{B}] \gtrsim \frac{\delta^{\alpha^*/(\alpha^*-1)} e^{-\alpha^* \pi}}{Q},$$

for

$$\begin{aligned}\beta' = &\sqrt{1 + (\sqrt{nm_1} + \sqrt{nm_2} + t)^2} \cdot \sqrt{2\sigma_1^2 nm_1 + 2\sigma^2 nm_2} \\ &+ (\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3},\end{aligned}$$

and where α^* is defined by $\alpha^* = 1 + \sqrt{\log_2(1/\delta)/(\pi \log_2 e)}$.

4 Zero-Knowledge Arguments of Knowledge

We now detail out the zero-knowledge arguments of knowledge (ZKAoK) that we use to instantiate the protocols from Section 5. Since we propose a construction over $\mathbb{Z}_q$ and another over structured lattices, we employ the frameworks from [YAZ+19] and [LNP22] respectively to tackle the relations to be proven. We first discuss some aspects of the former, and later explain in Section 4.3 how to use both frameworks to instantiate the necessary relations.

4.1 A Framework for Quadratic Relations over $\mathbb{Z}_q$

Our construction requires a proof system that handles exact quadratic relations, over $\mathbb{Z}_q$ for our first construction and another framework over R_q for our structured variant. Let us first focus on the former. To handle such relations, we could have used Stern-like protocols but this would only reach constant soundness error, thus implying a large number of repetitions and hence bad performance. Additionally, the decomposition-extension methods used in the original scheme [LLM+16] make the relation to be proven much larger. To circumvent these two shortcomings, we instead use the more recent framework by Yang et al. [YAZ+19]. It combines the perks of Stern-like ZKAoK and Fiat-Shamir with Aborts ZKAoK to reach a framework with standard soundness and inverse polynomial soundness error. This requires fewer iterations as a result. More precisely, the framework of [YAZ+19] provides a ZKAoK for the relation

$$\begin{aligned}\mathcal{R}^* = \{((\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}); \mathbf{x}) \in & (\mathbb{Z}_q^{k \times L_{\mathbf{x}}} \times \mathbb{Z}_q^{k} \times ([L_{\mathbf{x}}]^3)^{L_{\mathcal{M}}}) \times \mathbb{Z}_q^{L_{\mathbf{x}}} : \ \overline{\mathbf{A}}\mathbf{x} = \mathbf{y} \bmod q \\ & \wedge \forall (h, i, j) \in \mathcal{M}, \mathbf{x}[h] = \mathbf{x}[i] \cdot \mathbf{x}[j] \bmod q\}.\end{aligned}$$

This relation can be used to prove that the witness vector is short, which we need for our verification equation for example. Concretely, any witness $x \in \mathbb{Z}_q$ that we need to prove smaller than some bound B is decomposed as $x_1, \ldots, x_\ell$, where $\ell = \lceil \log_2 B \rceil$, which are proved binary using the quadratic relation $x_i^2 = x_i \bmod q$. The downside of this approach is that it adds ℓ witnesses for each short element, which quickly becomes cumbersome. To address this issue, the authors of [YAZ+19] introduced a so-called *fast mode* that significantly reduces the size of the witness. We describe such a mode in Section 4.2 but also show that its analysis in [YAZ+19] is not entirely correct and thus provide a more thorough one. We also propose additional optimizations in Appendix E.

4.2 Zero-Knowledge Fast Mode Revisited

As explained above, the decomposition technique entails a $(\ell+1)$-fold increase of the witness, which is prohibitive for high-dimensional vectors. This has led the authors of [YAZ+19] to sketch a so-called *fast mode* to obtain drastic efficiency gains in this case. The idea is to relax the zero-knowledge argument, thus introducing a soundness gap, and prove knowledge of a solution $\mathbf{w}'$ of $\mathbf{P}\mathbf{w}' = \mathbf{v} \bmod q$ such that $\mathbf{w}'$ is nB-bounded instead of B-bounded, where n is the dimension of $\mathbf{w}$. More precisely, they consider the following relation

$$\begin{aligned}\mathcal{R}' = \{((\mathbf{P}, \mathbf{v}, \mathbf{H}, \mathbf{c}), (\mathbf{w}, \mathbf{u}, \mathbf{r})) \in & (\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m} \times [0,1]^{\lambda \times n} \times \mathtt{C}) \times (\mathbb{Z}_q^{n} \times [0, nB]^{\lambda} \times \mathtt{R}) : \\ & \mathbf{P}\mathbf{w} = \mathbf{v} \bmod q \wedge \mathbf{H}\mathbf{w} - \mathbf{u} = \mathbf{0} \bmod q \wedge \mathbf{c} = \mathtt{Commit}(\mathbf{w}; \mathbf{r})\}\end{aligned}$$

The point is that the prover now only has to prove a bound on the λ elements of $\mathbf{u}$ instead of the n elements from $\mathbf{w}$, which is very interesting when $\lambda \ll n$, a condition easily met in practice. The authors argue that, if one knows a witness $(\mathbf{w}, \mathbf{u}, \mathbf{r})$ satisfying $\mathcal{R}'$, it ensures that $\mathbf{w}$ is in $[0, nB]^n$, except with negligible probability over the randomness of $\mathbf{H}$. We provide a simple counter-example to the above. For example, assume a prover knows $\mathbf{w} = [-1, 1, \ldots, 1]^T$ such that $\mathbf{Pw} = \mathbf{v} \bmod q$. We now consider $\mathbf{H}$ to be a random matrix whose entries are independently distributed according to $U(\{0, 1\})$. We denote by $\mathbf{h}_i$ the i-th row of $\mathbf{H}$ for $i \in [\lambda]$. For all $i \in [\lambda]$, we have $\mathbb{P}_{\mathbf{h}_i}[\mathbf{h}_i^T \mathbf{w} \in [0, nB]] = 1 - 2^{-n}$ by simply conditioning on the first coefficient of $\mathbf{h}_i$. It yields $\mathbb{P}_{\mathbf{H}}[\mathbf{Hw} \in [0, nB]^\lambda] = (1 - 2^{-n})^\lambda \geq 1 - \lambda 2^{-n}$. Since the *fast mode* is only relevant when $n \geq \lambda$, it holds that $\mathbf{Hw} \in [0, nB]^\lambda$ with overwhelming probability. This shows that $\mathcal{R}'$ cannot be used to prove that $\mathbf{w}$ has non-negative coefficients and thus for example invalidates the use of the fast mode in the e-cash use-case in [YAZ$^+$19].

Fortunately, a more thorough analysis shows that $\mathbf{Hw} \bmod q$ is in $[0, B]^\lambda$ implies that $\mathbf{w} \bmod q \in [-2B, 2B]^n$ with high probability, which would be sufficient in our case as we only need to prove bounds on the ℓ_∞ norm. However, we have so far only discussed of soundness. When it comes to correctness, we note that the choices made in [YAZ$^+$19] results in an unwieldy situation.

First, because one has to set an upper bound on $\mathbf{Hw}$ that will be satisfied with high probability for any $\mathbf{w}$ in $[-B, B]^n$. For a binary matrix $\mathbf{H}$, it seems hard to do much better than $[-nB, nB]^\lambda$ since we will be close to this bound for $\mathbf{w} = [B, \ldots, B]^T$, hence the factor n in the soudness gap mentioned above.

Second, because one cannot start the argument with $\mathbf{w} \in [-B, B]^n$ as it can lead to having $\mathbf{Hw}$ with negative coefficients. One must shift all the coefficients of $\mathbf{w}$ before running the protocol, but it results in a skewed statement on $\mathbf{w}$. Indeed, it would prove that $\mathbf{w} + B\mathbf{1}_n$ is in $[-2nB, 2nB]^n$ and therefore that $\mathbf{w} \in [-(2n+1)B, (2n-1)B]^n$, where $\mathbf{1}_n = [1 \ldots 1]^T \in \mathbb{Z}^n$.

For these reasons, we believe it is much more natural to sample the coefficients of $\mathbf{H}$ uniformly from $\{-1, 0, 1\}$. We prove below that $\mathbf{Hw} \bmod q$ is in $[-B, B]^\lambda$ still implies that $\mathbf{w} \bmod q \in [-2B, 2B]^n$, which avoids to shift the witness and thus the problem mentioned above. Moreover, such distribution of $\mathbf{H}$ allows us to derive much better upper bounds on $\mathbf{Hw}$ using for example an argument similar to the one of lemma 2.5. However, we do not study more thoroughly this general problem as we are able to derive sharp bounds for our specific use case (see remark 4.1 below).
More formally, let $\mathbf{H} \in [-1, 1]^{k \times n}$, with $k = \lambda / \log_2(9/5)$. The following lemma, proven in Appendix D, argues that $\mathbf{Hw} \bmod q \in [-B, B]^k$ implies $\mathbf{w} \bmod q \in [-2B, 2B]^n$ with overwhelming probability over the choice of $\mathbf{H}$.

Lemma 4.1. *Let $B \in \mathbb{Z}$ be such that $6B < q/2$. Let k be a positive integer. Let $\mathbf{w} \in \mathbb{Z}^n$ be a vector. Assuming that $\|\mathbf{w} \bmod q\|_\infty > 2B$, it then holds that $\mathbb{P}_{\mathbf{H} \hookleftarrow U([-1,1]^{k \times n})}[\|\mathbf{Hw} \bmod q\|_\infty \leq B] \leq (5/9)^k$.*

The fast mode that we consider now corresponds to the following relation, where B is chosen so that $\|\mathbf{Hw} \bmod q\|_\infty \leq B$ with overwhelming probability for an honest witness $\mathbf{w}$.

$$\begin{aligned}\mathcal{R}'' = \{ & ((\mathbf{P}, \mathbf{v}, \mathbf{H}, \mathbf{c}), (\mathbf{w}, \mathbf{u}, \mathbf{r})) \in (\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \times [-1, 1]^{k \times n} \times \mathtt{C}) \times \\ & (\mathbb{Z}_q^n \times [-B, B]^k \times \mathtt{R}) : \mathbf{P}\mathbf{w} = \mathbf{v} \bmod q \wedge \mathbf{H}\mathbf{w} - \mathbf{u} = \mathbf{0} \bmod q \wedge \mathbf{c} = \mathtt{Commit}(\mathbf{w}; \mathbf{r})\}\end{aligned}$$

Remark 4.1. For our relations, the vectors that we need to prove short are sampled from discrete Gaussian distributions. For example the vector $\mathbf{v}_1$ follows $\mathcal{D}_{\mathbb{Z}^{m_1}, \sigma_1}$. For a fixed $\mathbf{H} \in \{-1, 0, 1\}^{k \times m_1}$, the third statement of Lemma 2.3 yields that $\mathbb{P}_{\mathbf{v}_1}[|\langle \mathbf{v}_1, \mathbf{h}_i \rangle| \geq \sigma_1 t \sqrt{m_1}] \leq \mathbb{P}_{\mathbf{v}_1}[|\langle \mathbf{v}_1, \mathbf{h}_i \rangle| \geq \sigma_1 t \|\mathbf{h}_i\|_2] \leq 2e^{-\pi t^2}$, where $\mathbf{h}_i$ is the i-th row of $\mathbf{H}$ and the first inequality follows by event inclusion as $\|\mathbf{h}_i\|_2 \leq \sqrt{m_1}$. The union bound yields $\mathbb{P}_{\mathbf{v}_1}[\|\mathbf{H}\mathbf{v}\|_\infty \geq \sigma_1 t \sqrt{m_1}] \leq 2ke^{-\pi t^2}$, where $k = \lambda / \log_2(9/5)$ as per Lemma 4.1. Hence, taking $t = \log_2 \lambda$ gives that $\|\mathbf{H}\mathbf{v}\|_\infty \leq \sigma_1 \sqrt{m_1} \log_2 \lambda$ with overwhelming probability. This improves on the trivial bound $\sigma_1 m_1 \log_2 m_1$. By making sure that $2\sigma_1 \sqrt{m_1} \log_2 \lambda < (q-1)/2$, which is generally the case, we have no wrap-around modulo q in $\mathbf{H}\mathbf{v}_1$ and therefore $\|\mathbf{H}\mathbf{v}_1 \bmod q\|_\infty \leq \sigma_1 \sqrt{m_1} \log_2 \lambda$. The conditions of Lemma 4.1 allow one to choose $B = \sigma_1 \sqrt{m_1} \log_2 \lambda \ll q/12$. Then, proving that $\|\mathbf{H}\mathbf{v}_1 \bmod q\|_\infty \leq \sigma_1 \sqrt{m_1} \log_2 \lambda$ implies that $\|\mathbf{v}_1 \bmod q\|_\infty \leq 2\sigma_1 \sqrt{m_1} \log_2 \lambda$.

4.3 Zero-Knowledge Arguments and Relations

The zero-knowledge framework from [YAZ$^+$19] allows to prove quadratic relations over $\mathbb{Z}_q$. The protocols accompanying our signature that we present in Section 5 require a proof system to prove knowledge of a commitment opening, and to prove knowledge of a message-signature pair, which are both quadratic relations. At a high level, the commitment opening proof requires to prove a linear relation and that the witness is short. As explained when describing $\mathcal{R}^*$, the latter can be dealt with by decomposing each entry in a binary vector, and proving that the latter indeed has binary coefficients. Similarly, proving knowledge of $(\mathbf{m}, \tau, \mathbf{v})$ such that $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, (\tau, \mathbf{v}), \mathsf{pp}) = 1$, requires proving that some elements have small magnitude, and that $\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \tau\mathbf{G}\mathbf{v}_2 = \mathbf{u} + \mathbf{D}\mathbf{m} \bmod q$ which is quadratic because of the term $\tau\mathbf{v}_2$. Due to lack of space, we defer to Appendix F the details on how to use the framework of [YAZ$^+$19] to instantiate them.

We however give more details for our construction over structured lattices. Although the framework of [YAZ$^+$19] straightforwardly adapts to the ring or module setting, it results in relations of the form $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod qR$ and $\mathbf{x}[h] = \mathbf{x}[i]\mathbf{x}[j] \bmod qR$. In our case, we aim to prove that the witness is short (or binary for the message part) with respect to the coefficient embedding of R. Taking the example of the message, $\mathbf{m}[i] = \mathbf{m}[i]^2 \bmod qR$ does not imply that the coefficients of the polynomial $\mathbf{m}[i]$ are binary, but only that the number theoretic transform (NTT) of $\mathbf{m}[i]$ is a binary vector. A naive alternative would be to embed the entire relation into $\mathbb{Z}$ via the coefficient embedding and applying [YAZ$^+$19] in a non-structured way. This would indeed prove the desired relation but it would also ignore the underlying structure and all the optimizations that come with it. Instead, we use the very recent framework by Lyubashevsky, Nguyen and Plançon [LNP22], which generalizes the previous work of [BLS19] and [ENS20] used to obtain exact proofs. The advantage of this framework is that it provides a way to prove bounds on the ℓ_2 norm of the witness without resorting to bounds on the ℓ_∞ norm. As explained in [LNP22], this leads to proving tighter bounds on the ℓ_2 norm, and in a more efficient way as a result. We denote by σ_{-1} to be the automorphism of R_q that can be defined as $\sigma_{-1}(\sum_{i=0}^{n-1} r_i X^i) = r_0 - \sum_{i=1}^{n-1} r_i X^{n-i}$. Their proof system allows one to prove relations of the form

$$\begin{cases} \forall i \in [\rho], f_i(\mathbf{s}) = 0 \bmod qR & \forall i \in [v_e], \left\|\mathbf{E}_i^{(e)}\mathbf{s} - \mathbf{u}_i^{(e)}\right\|_2 \leq \beta_i^{(e)} \\ \forall i \in [\rho_{eval}], \widetilde{F}_i(\mathbf{s}) = 0 & \forall i \in [v_a], \left\|\mathbf{E}_i^{(a)}\mathbf{s} - \mathbf{u}_i^{(a)}\right\|_\infty \leq \beta_i^{(a)}, \end{cases}$$

where the f_i, F_i are quadratic functions in $\mathbf{s} = [\mathbf{s}_1^T, \sigma_{-1}(\mathbf{s}_1)^T]^T$ ($\mathbf{s}_1$ being the committed vector), and $\widetilde{F}_i(\mathbf{s})$ denotes the constant coefficient of the polynomial $F_i(\mathbf{s})$. The norm conditions with superscript (e) are proven exactly, while those with superscript (a) are proven approximately. We note for completeness that the considered automorphism is not necessarily σ_{-1}. We present here how our relations can be instantiated in their framework, which consists in describing the functions f_i, F_i and matrices and vectors for the norm conditions.

Let $q_1 < q$ be a prime integer such that $q_1 = 2k+1 \bmod 4k$, and define $q_\pi = q_1 q$ as the modulus of the proof system, which is different from the modulus of our signature. We take q_1 having the same splitting as q in R to ensure the invertibility of challenge differences in R_{q_π} as discussed in [LNP22, Sec. 2.3].

4.3.1 Proof of Commitment Opening.

Consider the relation

$$q_1(\mathbf{Ar}' + \mathbf{Dm}) = q_1\mathbf{c} \bmod q_\pi R \wedge \|\mathbf{r}'\|_2 \leq \sigma_3\sqrt{nm_1} =: \alpha_3 \wedge \mathbf{m} \in S_{bin}^{m_3},$$

where the private input is $\mathbf{r}', \mathbf{m}$ and the public input is $\mathbf{A}, \mathbf{D}, \mathbf{c}$. We multiply the linear equation by q_1 to work with the proof system modulus. We now instantiate this relation in the framework of [LNP22]. Using the notations of [LNP22], we define $\mathbf{s}_1 = [\mathbf{r}'|\mathbf{m}] \in R^{m_1+m_3}$ and $\mathbf{s} = [\mathbf{s}_1|\sigma_{-1}(\mathbf{s}_1)] \in R^{2(m_1+m_3)}$.

Quadratic Equations: Define $f_i(\mathbf{s}) = (\mathbf{e}_i^T[q_1\mathbf{A}|q_1\mathbf{D}|\mathbf{0}_{d\times m_1+m_3}])\cdot\mathbf{s} + (-\mathbf{e}_i^T q_1\mathbf{c})$ for all $i \in [d]$, where $\mathbf{e}_i$ is the zero vector with a 1 at position i. Then, proving $f_i(\mathbf{s}) = 0 \bmod q_\pi R$ for all $i \in [d]$ yields $q_1(\mathbf{Ar}' + \mathbf{Dm}) = q_1\mathbf{c} \bmod q_\pi R$.

Quadratic Evaluations: We define $r = \sum_{j\in[0,n-1]} X^j$. For all $i \in [m_3]$, define $F_i(\mathbf{s}) = \mathbf{s}^T\mathbf{E}_{2m_1+m_3+i,m_1+i}\mathbf{s} + (-r\mathbf{e}_{2m_1+m_3+i})^T\mathbf{s} = \sigma_{-1}(\mathbf{m}[i])(\mathbf{m}[i] - r)$, where $\mathbf{E}_{k,\ell}$ denotes the zero matrix with a 1 at position (k,ℓ). Then, proving $\widetilde{F}_i(\mathbf{s}) = 0$ for all $i \in [m_3]$ implies $\mathbf{m} \in S_{bin}^{m_3}$. This relies on the fact that for $m \in R$, the constant coefficient of $\sigma_{-1}(m)(m-r)$ is $\langle\theta(m), \theta(m) - \mathbf{1}_n\rangle$. Then, proving that this inner product is 0 over $\mathbb{Z}$ is equivalent to proving that $\theta(m) \in \{0,1\}^n$, i.e., $m \in S_{bin}$.

Norm Conditions: We define $\mathbf{E}^{(e)} = [\mathbf{I}_{m_1}|\mathbf{0}_{m_1\times m_1+2m_3}]$, $\mathbf{u}^{(e)} = \mathbf{0}_{m_1}$, and $\beta^{(e)} = \alpha_3$. Then $\left\|\mathbf{E}^{(e)}\mathbf{s} - \mathbf{u}^{(e)}\right\|_2 \leq \beta^{(e)}$ is equivalent to $\|\mathbf{r}'\|_2 \leq \alpha_3$.

Remark 4.2. The above aims at proving the relation exactly. However, we note that the commitment scheme employed in [LNP22] already contains a part $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$. By setting the public matrices $\mathbf{A}_1, \mathbf{A}_2$ as $\mathbf{A}, \mathbf{D}$ respectively, $\mathbf{s}_2 = \mathbf{r}'$ which is chosen from a Gaussian distribution, and $\mathbf{s}_1 = \mathbf{m}$, we can directly use the protocol of [LNP22, Fig. 8]. We simply have to set $\|\mathbf{s}_1\|_2 \leq \sqrt{nm_3} =: \alpha$, and the quadratic evaluations as above to prove (exactly) that $\mathbf{s}_1 = \mathbf{m}$ is indeed in $S_{bin}^{m_3}$. It then proves the correct statement but with a soundness gap on the norm of $\mathbf{r}'$.

4.3.2 Proof of Message-Signature Pair Possession.

Consider the relation

$$q_1(\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \tau\mathbf{G}\mathbf{v}_2 - \mathbf{D}\mathbf{m}) = q_1\mathbf{u} \bmod q_\pi R$$
$$\text{with } \|\mathbf{v}\|_2 \leq \sqrt{\sigma_1^2 nm_1 + \sigma^2 nm_2} =: \alpha \wedge \mathbf{m} \in S_{bin}^{m_3} \wedge \tau \in \mathcal{T}_w,$$

where the private input is $\tau, \mathbf{v} = [\mathbf{v}_1^T | \mathbf{v}_2^T]^T, \mathbf{m}$ and the public input is composed of $\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{G}, \mathbf{u}$. We define $\mathbf{s}_1 = [\mathbf{v}_1 | \mathbf{v}_2 | \mathbf{m} | \tau] \in R^{m_1+m_2+m_3+1}$ and $\mathbf{s} = [\mathbf{s}_1 | \sigma_{-1}(\mathbf{s}_1)] \in R^{2(m_1+m_2+m_3+1)}$.

Quadratic Equations: We define $\mathbf{A}' = q_1[\mathbf{A} | -\mathbf{B} | -\mathbf{D} | \mathbf{0}_{d \times m_1+m_2+m_3+2}]$, and for all $i \in [d]$, we define

$$\mathbf{G}_i = q_1 \begin{bmatrix} \mathbf{0}_{(m_1+m_2+m_3) \times 2(m_1+m_2+m_3+1)} \\ \mathbf{0}_{1 \times m_1} \ \mathbf{e}_i^T\mathbf{G} \ \ \mathbf{0}_{1 \times m_1+m_2+2(m_3+1)} \\ \mathbf{0}_{(m_1+m_2+m_3+1) \times 2(m_1+m_2+m_3+1)} \end{bmatrix}.$$

Then, for all $i \in [d]$, define $f_i(\mathbf{s}) = \mathbf{s}^T\mathbf{G}_i\mathbf{s} + (\mathbf{e}_i^T\mathbf{A}')\mathbf{s} + (-q_1\mathbf{e}_i^T\mathbf{u})$. Proving $f_i(\mathbf{s}) = 0 \bmod q_\pi R$ for all $i \in [d]$ yields $q_1(\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \tau\mathbf{G}\mathbf{v}_2 - \mathbf{D}\mathbf{m}) = q_1\mathbf{u} \bmod q_\pi R$.

Quadratic Evaluations: We define $r = \sum_{j \in [0,n-1]} X^j$. For all $i \in [m_3+1]$, define $F_i(\mathbf{s}) = \mathbf{s}^T\mathbf{E}_{2(m_1+m_2)+m_3+1+i, m_1+m_2+i}\mathbf{s} + (-r\mathbf{e}_{2(m_1+m_2)+m_3+1+i})^T\mathbf{s}$. We also define $F_{m_3+2}(\mathbf{s}) = \mathbf{s}^T\mathbf{E}_{2(m_1+m_2+m_3+1), m_1+m_2+m_3+1}\mathbf{s} - w = \sigma_{-1}(\tau)\tau - w$. Proving $\widetilde{F}_i(\mathbf{s}) = 0$ for $i \in [m_3]$ is equivalent to $\mathbf{m} \in S_{bin}^{m_3}$ as before. Then, showing $\widetilde{F}_{m_3+1}(\mathbf{s}) = 0$ proves $\tau \in S_{bin}$, while $\widetilde{F}_{m_3+2}(\mathbf{s}) = 0$ proves that $\|\tau\|_2^2 = \langle\theta(\tau), \theta(\tau)\rangle = w$, thus giving $\tau \in \mathcal{T}_w$.

Norm Conditions: We define $\mathbf{E}^{(e)} = [\mathbf{I}_{m_1+m_2} | \mathbf{0}_{m_1+m_2 \times m_1+m_2+2(m_3+1)}]$, $\mathbf{u}^{(e)} = \mathbf{0}_{m_1+m_2}$, and $\beta^{(e)} = \alpha$. Then $\left\|\mathbf{E}^{(e)}\mathbf{s} - \mathbf{u}^{(e)}\right\|_2 \leq \beta^{(e)}$ proves $\|\mathbf{v}\|_2 \leq \alpha$.

5 Privacy-Preserving Protocols and Anonymous Credentials

The very purpose of a signature scheme with efficient protocols (SEP) is to be used as a building block for privacy-preserving primitives such as group signature, anonymous credentials or e-cash. For such applications, one usually needs (1) to get a signature on committed (hidden) messages and (2) to prove knowledge of a signature, without revealing it. This has led previous papers, e.g., [CL04,PS16,LLM+16], providing such types of signatures to describe specific protocols addressing those needs. We here follow the same approach. More specifically, we give a first protocol in Section 5.1 which allows a signer to obliviously sign a message, by only knowing a commitment to the message. The second protocol, presented in Section 5.2, enables a user to prove the possession of a message-signature pair, where the signature has been obtained by the oblivious signing protocol. As in previous works, we do not identify any properties expected from such protocols nor prove any results regarding their security. As this might look unconventional, we need to recall a few facts about SEPs and their use in privacy-preserving applications.

The use of signature schemes in the latter applications can be done based on formal generic frameworks, e.g., [BSZ05] for group signature or [FBS19] for e-cash, or on some rather common heuristics, e.g., for anonymous credentials [CL04]. In all cases, the point is that, in theory, no specific property is expected from the signatures beyond EUF-CMA security. Typically, [BSZ05] and [FBS19] consider standard digital signature schemes for their framework.

However, in practice, the use of any digital signature is likely to lead to a totally impractical construction because of the difficult interactions between general-purpose signatures and the other building blocks such as zero-knowledge proofs. This is where SEPs prove handy. They are specifically designed to smoothly interact with the other building blocks so as to optimize the efficiency of the resulting construction.

In this context, defining security notions that such protocols should achieve would be meaningless as no such formal properties are expected by the constructions using them. Worse, this is likely to lead to unnecessary complications as it is difficult to define a relevant security model for SEPs. Typically, an SEP allows one to get a signature on hidden messages and then to prove knowledge of the message-signature pair. How to define a relevant security model in this context? Unforgeability indeed means the inability to produce a signature on new messages but here we do not know the messages requested by the adversary to the signing oracle and we do not know which message-signature pairs it is proving knowledge of. In other words, we cannot decide if the adversary won.

Libert et al [LLM+16] circumvents this issue by forcing the user to provide an encryption of the messages in the blind issuance process. This does not address the problem of formalizing the properties expected from the protocols (1) and (2) of SEPs (and indeed [LLM+16] does not define such properties) but this enables to provide some results regarding security as a reduction can recover all the messages it has signed (by decrypting the ciphertexts) and thus decide when a forgery occurs. Besides being unconventional (this led [LLM+16] to prove "security" of the protocols without defining what "security" means), this approach complicates the protocols by adding this encryption step that is not necessary in most applications using such signatures. Indeed, in concrete applications, this problem is usually solved by other means. For example, in e-cash systems, "forgeries" can be detected by comparing the amount of withdrawn coins with the one of spent coins. In group signatures, there is an opening procedure that allows to trace back a group signature to a group member. This enables to detect forgeries as the latter will be involved in group signatures that cannot be opened to anyone.

To sum up, SEPs constitute an informal subclass of digital signatures designed for privacy-preserving applications. Defining specific security properties for the protocols associated with SEPs is not necessary for such applications and artificially increases complexity. In accordance with previous works, we therefore do not consider such security properties.

However, to demonstrate how an SEP can easily be plugged in a privacy-preserving construction and how security is concretely managed in this case, we provide in Section 5.3 the description of an anonymous credentials system based on our SEP scheme.

In this section, we present the protocols for the construction over structured lattices, but they can be naturally adapted for the construction over $\mathbb{Z}_q$. We thus use the zero-knowledge arguments presented in Section 4.3 using the framework from [LNP22]. One would instead use the framework of [YAZ+19] for the relations over $\mathbb{Z}_q$, which are detailed in Appendix F.

5.1 Oblivious Signing Protocol

We present here our first protocol between a signer S and a user U. The user U is interacting with S in order to obtain a signature $(\tau, \mathbf{v})$ on a message $\mathbf{m}$, by only providing S with a commitment $\mathbf{c}$ to the message $\mathbf{m}$. We assume that Algorithms 3.5 and 3.6 have been run prior to entering the protocol but with some slight modifications that we detail below. First, instead of choosing σ_2 as in Algorithm 3.5, it chooses $\sigma_3 \geq \sqrt{2}\eta_\varepsilon(R^{m_1})$ and then

$$\sigma_4 \geq \max\left(\sqrt{((\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3} + \sigma_3\sqrt{nm_1})^2 - \sigma^2}, \sqrt{2}\eta_\varepsilon(R^{m_1})\right).$$

It then re-defines $\sigma_2 = \sqrt{\sigma_3^2 + \sigma_4^2}$ and $\sigma_1 = \sqrt{\sigma^2 + \sigma_2^2}$. The new widths σ_3, σ_4 are also included in pp in addition to $\sigma, \sigma_1, \sigma_2$. We explain this change in Remark 5.1. Second, as we use the public key matrix $\mathbf{A}$ as part of the commitment matrices, we must ensure that it cannot be tempered with by the attacker. As such, we generate $\mathbf{A}$ as the hash of a public string. In the random oracle model, the matrix can be assumed to follow the prescribed uniform distribution over $R_q^{d \times m_1}$.

Algorithm 5.1: OblSign (Oblivious Signing Interactive Protocol)

```
Input: Signer $S$ with sk, pk, pp, st, and a user $U$ with $\mathbf{m} \in S_{bin}^{m_3}$ and pk, pp.
    User $U$.
 1. $\mathbf{r}' \hookleftarrow \mathcal{D}_{R^{m_1},\sigma_3}$.                ▷ $\sigma_3 \geq \sqrt{2}\eta_\varepsilon(R^{m_1})$
 2. $\mathbf{c} \leftarrow \mathbf{A}\mathbf{r}' + \mathbf{D}\mathbf{m} \bmod qR$.
 3. Send $\mathbf{c}$ to $S$.
    User $U \longleftrightarrow$ Signer $S$.
 4. Interactive zero-knowledge argument between $U$ and $S$, where $U$ proves that $\mathbf{c}$
    is commitment to $\mathbf{m}$ with randomness $\mathbf{r}'$. If $S$ is not convinced, the protocol
    aborts.
    Signer $S$.
 5. $\mathbf{r}'' \hookleftarrow \mathcal{D}_{R^{m_1},\sigma_4}$.
```

```
$\mathbf{c}' \leftarrow \mathbf{c} + \mathbf{A}\mathbf{r}'' \bmod qR$.
$\tau \leftarrow F(\mathsf{st})$.                                  ▷ $\tau \in \mathcal{T}_w$
$\mathbf{v}' \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \tau\mathbf{I}_d, \mathbf{u} + \mathbf{c}', \sigma) - [\mathbf{r}''^T|\mathbf{0}]^T$.
Send $(\tau, \mathbf{v}')$ to $U$.
$\mathsf{st} \leftarrow \mathsf{st} + 1$
User $U$.
$\mathbf{v} \leftarrow \mathbf{v}' - [\mathbf{r}'^T|\mathbf{0}]^T$.
if $\mathsf{Verify}(\mathsf{pk}; \mathbf{m}; (\tau, \mathbf{v}); \mathsf{pp}) = 1$, then return $(\tau, \mathbf{v})$.          ▷ Algorithm 3.8
else return $\perp$
```

Remark 5.1. Notice that Algorithm 5.1 does not exactly rely on the signature scheme of Section 3. This is because the signer S also contributes to the randomness of the commitment to the message $\mathbf{m}$ via $\mathbf{r}''$. If the randomness came only from the user U, the signer, who is embodied by the SIS adversary in the security proofs, would have no control over the randomness part of the signing query. In the proof of Lemma 3.4 (and Lemma 3.5 for the i-th query with $i \neq i^+$), the randomness $\mathbf{r}$ is legitimately sampled from $\mathcal{D}_{R^{m_1},\sigma_2}$. As such, it could instead be sampled as $\mathbf{r}' + \mathbf{r}''$ with $\mathbf{r}' \hookleftarrow \mathcal{D}_{R^{m_1},\sigma_3}$ sampled by the forger $\mathcal{A}$, and $\mathbf{r}'' \hookleftarrow \mathcal{D}_{R^{m_1},\sigma_4}$ sampled by the SIS adversary, thus matching with Algorithm 5.1. This would restrict $\sigma_2 = \sqrt{\sigma_3^2 + \sigma_4^2}$. If $\sigma_3, \sigma_4 \geq \sqrt{2}\eta_\varepsilon(R^{m_1})$, Lemma 2.2 guarantees that $\mathbf{r}' + \mathbf{r}''$ is $7\varepsilon/4$-close to $\mathcal{D}_{R^{m_1},\sigma_2}$ as required. However, when dealing with the i^+-th query in Lemma 3.5, the SIS adversary needs to control part of the randomness. At this step of the proof, $\mathbf{r}_0$ would be distributed according to $\mathcal{D}_{R^{m_1},\sigma_4}$, and it would construct $\mathbf{v}_1'^{(i^+)} = \mathbf{v}_1 - (\mathbf{r}_0 - \mathbf{U}\mathbf{m}^{(i^+)} - \mathbf{r}'^{(i^+)})$ with $\mathbf{r}'^{(i^+)}$ sampled from $\mathcal{D}_{R^{m_1},\sigma_3}$ by the forger $\mathcal{A}$. The rest remains the same, but this modification introduces the condition $\sqrt{\sigma^2 + \sigma_4^2} \geq \alpha + \sigma_3\sqrt{m_1}$, where $\alpha = (\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3}$. It yields $\sigma_2 \geq \sqrt{(\alpha + \sigma_3\sqrt{nm_1})^2 + \sigma_3^2 - \sigma^2}$, leading to $\sigma_1 = \sqrt{\sigma^2 + \sigma_2^2} \geq \sqrt{(\alpha + \sigma_3\sqrt{nm_1})^2 + \sigma_3^2}$ instead of just α. In most applications, m_3 is much larger than σ_3 and it thus only entails a mild increase of σ_1.

5.2 Message-Signature Pair Possession Protocol

The second protocol provides a user, who obtained a certificate $\mathsf{sig} = (\tau, \mathbf{v})$ on a message $\mathbf{m}$, with the ability to prove possession of this valid message-signature pair. For that, they only have to prove that $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, (\tau, \mathbf{v}), \mathsf{pp}) = 1$ without revealing neither $\mathbf{m}$ nor $(\tau, \mathbf{v})$. The protocol of Algorithm 5.2 thus simply consists in using the ZKAoK presented in Section 4.3 to prove this relation. The proof can be made non-interactive in the random oracle model using the Fiat-Shamir transform.

Algorithm 5.2: Prove (Message-Signature Pair Possession)

```
Input: User $U$ with $\mathsf{pk}, \mathsf{pp}, \mathbf{m}, (\tau, \mathbf{v})$, and a verifier $V$ with $\mathsf{pk}, \mathsf{pp}$.
    User $U \longleftrightarrow$ Verifier $V$.
    Interactive zero-knowledge argument between $U$ and $V$, where $U$ proves knowledge of $(\mathbf{m}; (\tau, \mathbf{v}))$ such that $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, (\tau, \mathbf{v}), \mathsf{pp}) = 1$.
```

5.3 Application to Anonymous Credentials

Anonymous credentials (AC), a.k.a. attribute-based credentials, is a generic term covering a wide spectrum of privacy-preserving systems considering essentially two main use-cases. One where a user interacts with an organization to get a signature on potentially concealed attributes, and one where this user will prove possession of this signature on his attributes while limiting leakage to some threshold depending on the concrete applications. For example, one may agree to reveal some attributes but wants to retain unlinkability of showings, which implies to hide the signature. We refer to [FHS19] for a discussion on the different features of such systems. In all cases, one can note that our two protocols above readily address those needs. To demonstrate that, we formally describe the anonymous credentials system resulting from our SEP construction and prove that it satisfies the security model introduced in [FHS19]. This model is recalled in Appendix C.1 for completeness but, in a few words, anonymous credentials is defined by two Keygen algorithms, one (OKeyGen) for the organization issuing credentials and one (UKeyGen) for the user, along with two interactive protocols, one (Issue) run by the organization and the user who wants to obtain a certificate and one (Show) run by some user and some verifier to check the validity of the claimed attributes. From the security standpoint, two properties are expected: anonymity and unforgeability. The former informally requires that Show does not leak more information than necessary, i.e., the set of disclosed attributes. The latter requires that no user can claim a credential on some attributes unless it has personally received a certificate from the organization. This in particular implies that nobody can present a credential that they do not own.

The OKeyGen and UKeyGen algorithms and the (Issue) and (Show) protocols based on our SEP construction are presented below. It gives, to our knowledge, the first lattice-based anonymous credential system. The algorithm Setup (Alg. 3.5) is modified so that $m_3 = m_s + m_3'$ where $m_s = 2d$ and $m_3' \cdot n$ is the maximal total bitsize of the attributes. We consider a system with ℓ attributes, where $h_i \cdot n$ is the bitsize of the i-th attribute $\mathbf{m}_i$ which means $m_3' = \sum_{i \in [\ell]} h_i$. The commitment matrix $\mathbf{D}$ is decomposed into $\mathbf{D} = [\mathbf{D}_s | \mathbf{D}_1 | \ldots | \mathbf{D}_k]$ where $\mathbf{D}_s \in R_q^{n \times m_s}$ and $\mathbf{D}_i \in R_q^{n \times h_i}$.

Algorithm 5.3: OKeyGen

```
Input: Public parameters $\mathsf{pp}$ as in Algorithm 3.5.
Output: $(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$.        ▷ Algorithm 3.6
```

Algorithm 5.4: UKeyGen

```
Input: Public parameters $\mathsf{pp}$ as in Algorithm 3.5.
  1. $\mathbf{s} \hookleftarrow U(S_{bin}^{m_s})$.
  2. $\mathbf{t} \leftarrow \mathbf{D}_s \mathbf{s} \bmod qR$.
Output: $(\mathsf{upk}, \mathsf{usk}) = (\mathbf{t}, \mathbf{s})$.
```

Algorithm 5.5: Issue (Credential Issuance Protocol)

```
Input: Organization $O$ with $\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st}$, and a user $U$ with $\mathbf{m} \in S_{bin}^{m_3'}$
and $\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}$.
      User $U$ $\longleftrightarrow$ Organization $O$.
```

1. Run the interactive protocol $\mathsf{OblSign}$ from Algorithm 5.1, where O plays the signer and with message $\widetilde{\mathbf{m}} = [\mathsf{usk}^T|\mathbf{m}^T]^T$. In this syntax, i.e., [FHS19], the signer knows $\mathbf{m}$ but not usk. Hence the ZKAoK is adapted to prove knowledge of short $(\mathbf{r}', \mathbf{s})$ such that $\mathbf{c} - \sum_i \mathbf{D}_i \mathbf{m}_i = \mathbf{A}\mathbf{r}' + \mathbf{D}_s \mathbf{s} \bmod qR$, and additionally that $\mathbf{D}_s \mathbf{s} = \mathsf{upk} \bmod qR$.

Algorithm 5.6: Show (Credential Showing Protocol)

Input: User U with $\mathsf{usk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}, (\tau, \mathbf{v}), \mathcal{I}$, and verifier V with $\mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i)_{i \in \mathcal{I}}$.

User $U \longleftrightarrow$ Verifier V.

1. Interactive zero-knowledge argument between U and V, where U proves knowledge of $(\mathbf{s}, (\mathbf{m}_i)_{i \notin \mathcal{I}}; (\tau, \mathbf{v}))$ such that $\mathsf{Verify}(\mathsf{pk}, \widetilde{\mathbf{m}}, (\tau, \mathbf{v}), \mathsf{pp}) = 1$.

Theorem 5.1. *The AC described in Algorithms 5.3, 5.4, 5.5 and 5.6 is correct, anonymous under the zero-knowledge property of the underlying ZKAoKs, and unforgeable under the hardness of Inhomogeneous* M-SIS*,* M-LWE*, the soundness of the ZKAoKs and the EUF-CMA security of our SEP.*

The proof of Theorem 5.1 is given in Appendix C.2 for lack of space. We also defer the detailed performance analysis in Appendix G. As an example, for 10 128-bit attributes with 4 disclosed ones, the proof size in the Show protocol is less than 730 KB.

Conclusion

In this paper, we have proposed a new signature scheme with efficient protocols which is several orders of magnitude more efficient than the current state-of-the-art [LLM+16]. This improvement was obtained by revisiting the latter construction in a systematic way, considering not only the signature scheme itself but also its interactions with the other components such as the commitment scheme and the zero-knowledge proofs. In the process, we have also rectified a problem with the fast mode of the zero-knowledge framework of [YAZ+19] and introduced some optimizations, which are of independent interest.

Our construction was designed to remain as generic as possible in order to be compatible with the broadest possible spectrum of applications. In particular, it can be instantiated in both standard lattices and structured ones so as to suit any lattice-based system. Despite this versatility, the size of a proof of knowledge of a message-signature pair, one of the core component of privacy-preserving systems, can be much lower than 1 MB, which should foster the development of practical post-quantum constructions in this area. We made a step in this direction by giving the first lattice-based anonymous credentials system.

Acknowledgments. This work has received a French government support managed by the National Research Agency in the ASTRID program, under the national project AMIRAL with reference ANR-21-ASTR-0016 and in the MobiS5 project, with reference ANR-18-CE-39-0019-02 MobiS5. It is also supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701).

References

AG11. S. Arora and R. Ge. New algorithms for learning in presence of errors. In *ICALP*, 2011.

Ajt96. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, 1996.

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.

BDGL16. A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, 2016.

BDL$^+$18. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, 2018.

BEP$^+$21. P. Bert, G. Eberhart, L. Prabel, A. Roux-Langlois, and M. Sabt. Implementation of lattice trapdoors on modules and applications. In *PQCrypto*, 2021.

BJRW23. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module learning with errors with short distributions. *J. Cryptol.*, 36:1, 2023.

BL07. E. Brickell and J. Li. Enhanced privacy id: a direct anonymous attestation scheme with enhanced revocation capabilities. In *WPES*, 2007.

BLR$^+$18. S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the rényi divergence rather than the statistical distance. *J. Cryptol.*, 2018.

BLS19. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO*, 2019.

Boy10. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC*, 2010.

BSZ05. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, 2005.

Cha85. D. Chaum. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In *EUROCRYPT*, 1985.

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.

DH76. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 1976.

DM14. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *CRYPTO*, 2014.

DORS08. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 2008.

dPLS18. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *CCS*, 2018.

ENS20. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT*, 2020.

FBS19. D. Pointcheval F. Bourse and O. Sanders. Divisible e-cash from constrained pseudo-random functions. In *ASIACRYPT*, 2019.

FHS19. G. Fuchsbauer, C. Hanser, and D. Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *J. Cryptol.*, 2019.

HILL99. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 1999.

Int16. Intel. A cost-effective foundation for end-to-end iot security, white paper. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/intel-epid-iot-security-white-paper.pdf, 2016.

ISO13a. ISO/IEC. ISO/IEC 18370-2:2016 information technology — security techniques — blind digital signatures — part 2: Discrete logarithm based mechanisms. https://www.iso.org/standard/62544.html, 2013.

ISO13b. ISO/IEC. ISO/IEC 20008-2:2013 information technology — security techniques — anonymous digital signatures — part 2: Mechanisms using a group public key. https://www.iso.org/standard/56916.html, 2013.

Laa15. T. Laarhoven. Search problems in cryptography: From fingerprinting to lattice sieving, 2015. http://www.thijs.com/docs/phd-final.pdf.

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.

LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *IACR Cryptol. ePrint Arch.*, page 284, 2022. Version dated from March 07th 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.

LPR13. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-lwe cryptography. In *EUROCRYPT*, 2013.

LS18. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT*, 2018.

LSS14. A. Langlois, D. Stehlé, and R. Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *EUROCRYPT*, 2014.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

MP12. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

MP13. D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO*, 2013.

MR07. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 2007.

Pei08. C. Peikert. Limits on the hardness of lattice problems in l_p norms. *Comput. Complex.*, 2008.

PS16. D. Pointcheval and O. Sanders. Short randomizable signatures. In *CT-RSA*, 2016.

R6́1. A. Rényi. On measures of entropy and information. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, 1961.

Reg05. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

TCG15. TCG. https://trustedcomputinggroup.org/authentication/, 2015.

Ver12. Roman Vershynin. *Introduction to the non-asymptotic analysis of random matrices*. Cambridge University Press, 2012.

YAZ$^+$19. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO*, 2019.

A Proof of Lemma 2.5

We recall here the definition of a sub-exponential random variable. We say that a random variable X is sub-exponential with parameters (ν, α) if for all $t \in (-1/\alpha, 1/\alpha)$, $\mathbb{E}[\exp(t(X - \mathbb{E}[X])))] \leq \exp(t^2\nu^2/2)$. We have that a sum of m independent sub-exponential random variables with the same parameters (ν, α) is sub-exponential with parameters $(\nu\sqrt{m}, \alpha)$. Finally, it holds that for a sub-exponential random variable with parameter (ν, α)

$$\forall r > 0, \mathbb{P}[X - \mathbb{E}[X] \geq r] \leq \begin{cases} e^{-r^2/(2\nu^2)} & \text{if } 0 < r < \nu^2/\alpha \\ e^{\nu^2/(2\alpha^2) - r/\alpha} & \text{if } r \geq \nu^2/\alpha. \end{cases}$$

Proof (of Lemma 2.5). Let $\mathbf{m} \in \{0,1\}^m$ be an arbitrary vector, and we denote by $k = \|\mathbf{m}\|_1$ the number of ones in the vector. We consider the random matrix $\mathbf{U}$ whose entries are independent and identically distributed according to $U([-1,1])$, and we denote by u_{ij} the random variable representing the (i,j)-th entry of $\mathbf{U}$. For clarity, we also denote by $\mathbf{u}_i^T$ the i-th row of $\mathbf{U}$. We know that each u_{ij} is sub-Gaussian with parameter $\sqrt{2/3}$, i.e.,

$$\forall t \in \mathbb{R}, \mathbb{E}[\exp(tu_{ij})] \leq \exp(t^2/3).$$

By independence of the entries, we directly obtain for all $i \in [n]$

$$\forall t \in \mathbb{R}, \mathbb{E}[\exp(t\mathbf{u}_i^T\mathbf{m})] \leq \exp(kt^2/3).$$

Hence, each $\mathbf{u}_i^T\mathbf{m}$ is sub-Gaussian with parameter $s = \sqrt{2k/3}$. We define the random variables $x_i = \mathbf{u}_i^T\mathbf{m}$, $y_i = x_i^2$ and we also define $\mu_i = \mathbb{E}[y_i]$. Since x_i is sub-Gaussian with parameter s, we can prove that

$$\forall p \geq 1, \mathbb{E}[|x_i|^p] \leq p(\sqrt{2}s)^p\Gamma(p/2),$$

where Γ is the Gamma function. In particular, we have $\mu_i \leq 2(\sqrt{2}s)^2\Gamma(1) = 4s^2 = 8k/3$. We then have

$$\begin{aligned}
\mathbb{E}[e^{t(y_i - \mu_i)}] &= 1 + t\mathbb{E}[y_i - \mu_i] + \sum_{p=2}^{\infty} t^p\mathbb{E}[(y_i - \mu_i)^p]/p! \\
&\leq 1 + \sum_{p=2}^{\infty} t^p\mathbb{E}[x_i^{2p}]/p! \\
&\leq 1 + \sum_{p=2}^{\infty} t^p(2p(\sqrt{2}s)^{2p}\Gamma(p))/p! \\
&= 1 + 2\sum_{p=2}^{\infty}(2s^2t)^p \\
&= 1 + 8s^4t^2/(1 - 2s^2t),
\end{aligned}$$

where we used the fact that $\Gamma(p) = (p-1)!$ and that we restrict $|t| < 1/(2s^2\beta)$ for some free variable $\beta \geq 1$. It thus follows that

$$\mathbb{E}[e^{t(y_i-\mu_i)}] \leq 1 + 8\beta s^4 t^2/(\beta-1) \leq \exp(16\beta s^4/(\beta-1) \cdot t^2/2).$$

Hence, $y_i - \mu_i$ is a centered sub-exponential with parameters $\nu = 4s^2\sqrt{\beta/(\beta-1)}$ and $\alpha = 2s^2\beta$. We then define $y = \sum_{i\in[\ell]} y_i$ and $\mu = \sum_{i\in[\ell]} \mu_i$. It thus holds that $y - \mu$ is a centered sub-exponential with parameters $\nu\sqrt{\ell}$ and α. Using the tail bound above for a sub-exponential distribution, we have that for all $0 < r < \nu^2\ell/\alpha = 16\ell k/(3(\beta-1))$ then

$$\mathbb{P}[y - \mu \geq r] \leq \exp(-r^2/(2\ell\nu^2)).$$

Since the y_i are identically distributed, we have that $\mu = \ell\mu_1 \leq 8\ell k/3$. And we also have $y = \|\mathbf{Um}\|_2^2$. We now set the parameters β and r so that

$$\mathbb{P}[\|\mathbf{Um}\|_2 \geq 2\sqrt{\ell m}] \leq 2^{-x}.$$

In particular, we set $\beta = 1/(1 - 8x/(\ell \log_2 e))$. Assuming $\ell \geq 10x/\log_2 e$ entails $\beta \in (1, 5]$. Also, we set β this way to have $\sqrt{2\beta/((\beta-1)\log_2 e)}\sqrt{x/\ell} = 1/2$. Then, we set $r = 8k/3 \cdot \sqrt{2\beta/((\beta-1)\log_2 e)}\sqrt{\ell x} = 4\ell k/3$. We indeed have $r \leq 16\ell k/(3(\beta-1)) = \ell\nu^2/\alpha$. The way we set r, we have $\exp(-r^2/(2\ell\nu^2)) = 2^{-x}$, and $r + \mu \leq 4\ell k/3 + 8\ell k/3 = 4\ell k$. Hence

$$\mathbb{P}[\|\mathbf{Um}\|_2 \geq 2\sqrt{\ell k}] \leq \mathbb{P}[\sqrt{y} \geq \sqrt{r+\mu}] \leq \exp(-r^2/(2\ell\nu^2)) = 2^{-x},$$

In the worst case, we have $k = m$ which yields the claim. □

B Security Proofs

B.1 Additional Preliminaries

Probabilities. We denote by Supp(P) the support of the probability distribution P. In addition to the statistical distance, we use another measure of closeness between two probability distributions, namely the *Rényi divergence* [Ŕ61] RD. The Rényi divergence was thoroughly studied for its use in cryptography by Bai et al. [BLR+18] as it shows to be a powerful alternative to the statistical distance.

Definition B.1. *Consider two discrete probability distributions P and Q over a countable set S such that $Supp(P) \subseteq Supp(Q)$. We define the* Rényi divergence of order $\alpha > 1$ *as*

$$\mathrm{RD}_\alpha(P\|Q) = \left(\sum_{x\in Supp(P)} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}}\right)^{\frac{1}{\alpha-1}}.$$

The two measures enjoy a probability preservation property, which are essential in proving our results.

Lemma B.1. *Let P, Q be two probability distributions with $Supp(P) \subseteq Supp(Q)$, and $E \subseteq Supp(Q)$ be an arbitrary event. Then, $P(E) \leq \Delta(P, Q) + Q(E)$, and $P(E)^{\frac{\alpha}{\alpha-1}} \leq \mathrm{RD}_\alpha(P\|Q) \cdot Q(E)$.*

In the security proofs, we need to compute the Rényi divergence between two shifted discrete Gaussian distributions. We use the following lemma.

Lemma B.2 ([LSS14, Lem. 4.2]). *Let $\mathcal{L}$ be a lattice of rank n, and $\mathbf{c} \in \mathbb{R}^n$. Let $\alpha > 1$. Then, for any $\sigma > 0$, it holds that*

1. $\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sigma}\|\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}}) \leq \exp(\alpha\pi\|\mathbf{c}\|_2^2/\sigma^2)$,
2. $\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}}\|\mathcal{D}_{\mathcal{L},\sigma}) \leq \exp(\alpha\pi\|\mathbf{c}\|_2^2/\sigma^2) \cdot \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\alpha/(\alpha-1)}$, *if* $\sigma \geq \eta_\varepsilon(\mathcal{L})$.

Finally, to ensure that the syndrome generated by the SIS challenger is correctly distributed, we need to argue that $\mathbf{A}'\mathbf{v}'$ is close to uniform for a Gaussian vector $\mathbf{v}'$. We thus use the result of [MP12] which argues that the smoothing parameter of $\mathcal{L}_q^\perp(\mathbf{A}')$ is small with high probability over the choice of $\mathbf{A}'$.

Lemma B.3 (Adapted from [MP12, Lem. 2.4]). *Let n and q be positive integers with q prime, and let $m \geq n\log_2 q + \log_2(2 + 2\varepsilon^{-1})$ for some $\varepsilon > 0$. Let $\sigma \geq 2\eta_\varepsilon(\mathbb{Z}^m)$. Then for any $\delta > 0$, it holds that $\Delta((\mathbf{A}, \mathbf{A}\mathbf{e} \bmod q), (\mathbf{A}, \mathbf{u})) \leq \delta + 2\varepsilon/\delta$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$, $\mathbf{e} \sim \mathcal{D}_{\mathbb{Z}^m,\sigma}$, and $\mathbf{u} \sim U(\mathbb{Z}_q^n)$.*

In particular, choosing $\varepsilon = \delta^2/2$, $m > n\log_2 q + 2 - 2\log_2\delta$, $\sigma \geq \omega(\sqrt{\log_2 m})$ leads to a statistical distance of at most 2δ. In our case, we apply it with $m = m_1 + m_2 \gg n\log_2 q + 2\lambda + 4$, yielding a statistical distance much smaller than $2^{-\lambda}$.

Signature Security Model. The most widely used notion of security for a signature scheme is the *Existential Unforgeability against Chosen Message Attacks* (EUF-CMA) security. This captures the fact that an attacker that can obtain signatures on messages of its choosing is incapable of forging a signature on a new message. We formally define it by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ in Figure B.1.
The adversary's advantage is defined as $\mathrm{Adv}[\mathcal{A}] = \mathbb{P}[\mathcal{A} \text{ wins}]$, where the probability is over all the random coins. We say that the scheme is EUF-CMA secure if for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $\mathrm{Adv}[\mathcal{A}]$ is negligible.

B.2 Proof of Lemma 3.1

Proof. Consider a PPT adversary $\mathcal{A}$ that produces Type I forgeries for the signature scheme with advantage δ. We now construct an adversary $\mathcal{B}$ that solves the $\mathrm{SIS}^{\infty,2}_{n,m_1,q,\beta}$ problem. The adversary $\mathcal{B}$ is given $[\overline{\mathbf{A}}|\mathbf{u}] \in \mathbb{Z}_q^{n\times m_1+1}$ as input and is asked to find $\mathbf{w} \in \mathcal{L}_q^\perp([\overline{\mathbf{A}}|\mathbf{u}])$ such that $0 < \|\mathbf{w}\|_\infty \leq \beta_\infty$ and $0 < \|\mathbf{w}\|_2 \leq \beta_2$.

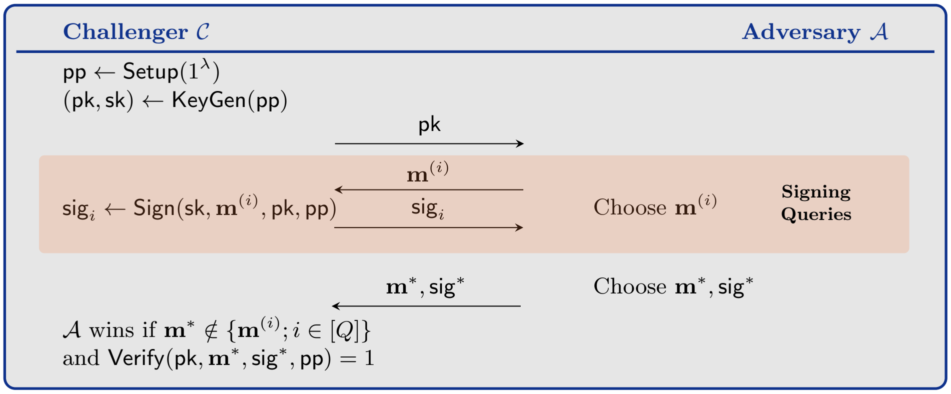
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
pk
$\mathbf{m}^{(i)}$
$\mathsf{sig}_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{m}^{(i)}, \mathsf{pk}, \mathsf{pp})$
sig_i
Choose $\mathbf{m}^{(i)}$
Signing Queries
$\mathbf{m}^*, \mathsf{sig}^*$
Choose $\mathbf{m}^*, \mathsf{sig}^*$
$\mathcal{A}$ wins if $\mathbf{m}^* \notin \{\mathbf{m}^{(i)}; i \in [Q]\}$
and $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}^*, \mathsf{sig}^*, \mathsf{pp}) = 1$

Fig. B.1. Existential Unforgeability against Chosen Message Attacks game

Setup Stage: $\mathcal{B}$ first generates the cryptographic material to give to $\mathcal{A}$. We assume that the parameters $\mathbf{g}, n, m_1, m_2, m_3, q, \sigma, \sigma_2, \sigma_1$. are already set. We also define $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}$. The adversary $\mathcal{B}$ first generates the tags $\tau^{(1)}, \ldots, \tau^{(Q)}$ that will be used for the signing queries of $\mathcal{A}$ by calling F and incrementing the state st. It also makes a guess $\overline{\tau} \hookleftarrow U(\mathcal{T} \setminus \{\tau^{(i)}; i \in [Q]\})$ on the tag that will be used in the adversary's forgery. In particular, we assume that $Q = \mathsf{poly}(\lambda)$ is the maximum number of signing queries that $\mathcal{A}$ is able to make.

Next, $\mathcal{B}$ samples $\mathbf{U}$ from $U([-1,1]^{m_1 \times m_3})$. It then randomizes $\overline{\mathbf{A}}$ to define $\mathbf{D} = \overline{\mathbf{A}}\mathbf{U} \bmod q$, and sets $\mathsf{pp} = (\mathbf{D}; \mathbf{g}; \lambda, n, m_1, m_2, m_3, q, \sigma, \sigma_2, \sigma_1)$. Then, $\mathcal{B}$ samples $\mathbf{R} \hookleftarrow U([-1,1]^{m_1 \times m_2})$ and defines $\mathbf{B} = \overline{\mathbf{A}}\mathbf{R} + \overline{\tau}\mathbf{G} \bmod q$. The adversary $\mathcal{B}$ then forms $\mathsf{pk} = (\overline{\mathbf{A}}, \mathbf{B}, \mathbf{u})$. From these matrices, we can define $\mathbf{A}_\tau$ for any tag $\tau \in \mathbb{Z}_{q'}$ by

$$\mathbf{A}_\tau = \left[\overline{\mathbf{A}} | \tau\mathbf{G} - \mathbf{B}\right] = \left[\overline{\mathbf{A}} | (\tau - \overline{\tau})\mathbf{G} - \overline{\mathbf{A}}\mathbf{R}\right], \tag{4}$$

Since $\overline{\tau}$ does not collide with the tags $\tau^{(1)}, \ldots, \tau^{(Q)}$ that will be used to answer the signing queries, we have $\tau^{(i)} - \overline{\tau} \in \mathbb{Z}_q^\times$ as q is prime. The matrices $\mathbf{A}_{\tau^{(i)}}$ thus have the adequate form to sample preimages using the trapdoor-based algorithms from [MP12]. Finally, $\mathcal{B}$ sends $(\mathsf{pk}, \mathsf{pp})$ to $\mathcal{A}$.

Query Stage: At the i-th signature query, $\mathcal{A}$ provides $\mathcal{B}$ with a message $\mathbf{m}^{(i)} \in \{0,1\}^{m_3}$. $\mathcal{B}$ can then faithfully run Algorithm 3.3 using the carefully crafted key material, and the tag $\tau^{(i)}$. More precisely, it computes $\mathbf{A}_{\tau^{(i)}}$ using Equation (4), as well as the message commitment $\mathbf{c} = \overline{\mathbf{A}}\mathbf{r}^{(i)} + \mathbf{D}\mathbf{m}^{(i)} \bmod q$ for a fresh randomness $\mathbf{r}^{(i)} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{m_1}, \sigma_2}$. As discussed, we can still use the $\mathbf{G}$-trapdoor $\mathbf{R}$ to sample preimages, allowing $\mathcal{B}$ to compute

$$\mathbf{v}^{(i)} = \mathsf{SampleD}(\mathbf{R}, \mathbf{A}, (\tau^{(i)} - \overline{\tau})\mathbf{I}_n, \mathbf{u} + \mathbf{c}, \sigma) - \begin{bmatrix} \mathbf{r}^{(i)} \\ \mathbf{0}_{m_2} \end{bmatrix}.$$

Note that $\mathbf{v}^{(i)}$ is correctly distributed and passes verification (with overwhelming probability by Lemma 2.2 and 2.3). The signature given to $\mathcal{A}$ is $\mathsf{sig}_i = (\tau^{(i)}, \mathbf{v}^{(i)})$.

Forgery Stage: After at most Q queries, the adversary returns a forgery $\mathsf{sig}^* = (\tau^*, \mathbf{v}^*)$ on a new message $\mathbf{m}^*$ that passes verification. If $\mathcal{A}$ fails to produce such a forgery, $\mathcal{B}$ aborts. We call this event Abort_1. We now condition on $\neg\mathsf{Abort}_1$. At this point, $\mathcal{B}$ aborts if $\tau^* \neq \overline{\tau}$. We call this event Abort_2 and further condition on $\neg\mathsf{Abort}_2$. Then, the guess was correct and therefore the contribution of $\mathbf{G}$ in $\mathbf{A}_{\tau^*}$ vanishes. Since the forgery passes verification we have $\mathbf{A}_{\tau^*}\mathbf{v}^* = \mathbf{u} + \mathbf{D}\mathbf{m}^* \bmod q$. Using the definition of the cryptographic material from the setup stage, it can be written as

$$\left[\overline{\mathbf{A}} | -\overline{\mathbf{A}}\mathbf{R}\right] \mathbf{v}^* = \mathbf{u} + \overline{\mathbf{A}}\mathbf{U}\mathbf{m}^* \bmod q.$$

This means that

$$\mathbf{w} = \begin{bmatrix} [\mathbf{I}_{m_1} | -\mathbf{R}]\mathbf{v}^* - \mathbf{U}\mathbf{m}^* \\ -1 \end{bmatrix} \in \mathbb{Z}^{m_1+1}$$

is in $\mathcal{L}_q^{\perp}([\overline{\mathbf{A}}|\mathbf{u}])$. The adversary $\mathcal{B}$ thus returns $\mathbf{w}$ as a solution for $\mathrm{SIS}^{\infty,2}_{n,m_1+1,q,\beta_\infty,\beta_2}$.

Advantage: We now analyze the advantage of $\mathcal{B}$. We first look at the distribution of $(\mathsf{pk}, \mathsf{pp})$. Since $m_1 \log_2 3 \geq n \log_2 q + f(\lambda)$, it holds by Lemma 2.1 that

$$\begin{cases} \Delta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{R} \bmod q), (\overline{\mathbf{A}}, U(\mathbb{Z}_q^{n\times m_2}))) \leq \frac{m_2}{2}\sqrt{\frac{q^n}{3^{m_1}}} \leq m_2 2^{-f(\lambda)/2-1}, \\ \Delta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{U} \bmod q), (\overline{\mathbf{A}}, U(\mathbb{Z}_q^{n\times m_3}))) \leq \frac{m_3}{2}\sqrt{\frac{q^n}{3^{m_1}}} \leq m_3 2^{-f(\lambda)/2-1}, \end{cases}$$

Additionally, since $\overline{\mathbf{A}}$, $\mathbf{R}$ are independent of $\overline{\tau}\mathbf{G}$, it holds that $\Delta(\mathbf{B}, \overline{\mathbf{A}}\mathbf{R}) \leq m_2 2^{-f(\lambda)/2}$ (by the triangle inequality). The signatures that are given to $\mathcal{A}$ in the query stage are distributed according to the legitimate distribution. This means that

$$\mathbb{P}[\neg\mathsf{Abort}_1] \geq \delta - \mathsf{negl}(\lambda). \tag{5}$$

As the guess $\overline{\tau}$ is independent of $\mathcal{A}'s$ view, we directly have

$$\mathbb{P}[\neg\mathsf{Abort}_2|\neg\mathsf{Abort}_1] = \frac{1}{|\mathcal{T}| - Q}. \tag{6}$$

We now analyze the solution provided by $\mathcal{B}$. We have to show it is non-zero and have ℓ_∞ norm at most β. Since the last coefficient of $\mathbf{w}$ is -1, we directly get that $\mathbf{w} \neq \mathbf{0}$. Then, by decomposing $\mathbf{v}^*$ into $[{\mathbf{v}_1^*}^T|{\mathbf{v}_2^*}^T]^T$, with $\mathbf{v}_1^* \in \mathbb{Z}^{m_1}$ and $\mathbf{v}_2^* \in \mathbb{Z}^{m_2}$, we have

$$\begin{aligned} \|\mathbf{w}\|_\infty &\leq \|\mathbf{v}_1^*\|_\infty + m_2\|\mathbf{R}\|_{\max}\|\mathbf{v}_2^*\|_\infty + m_3\|\mathbf{U}\|_{\max}\|\mathbf{m}^*\|_\infty \\ &\leq \sigma_1 \log_2 m_1 + m_2 \cdot \sigma \log_2 m_2 + m_3 \\ &= \beta_\infty. \end{aligned}$$

Now, since $\mathbf{v}_1^*, \mathbf{v}_2^*$ correspond to the forgery that passes verification, we only know their ℓ_∞ norm. In particular, we cannot apply the Gaussian tail bound to determine their ℓ_2 norm. Therefore, we can at best have $\|\mathbf{v}_1^*\|_2 \leq \sigma_1\sqrt{m_1}\log_2 m_1$

and $\|\mathbf{v}_2^*\|_2 \leq \sigma\sqrt{m_2}\log_2 m_2$. Also, note that the spectral norm of $[\mathbf{I}_{m_1}| - \mathbf{R}]$ is exactly $\sqrt{1 + \|\mathbf{R}\|_2^2}$. It follows that

$$\begin{aligned}
\|\mathbf{w}\|_2 &\leq \sqrt{1 + (\|[\mathbf{I}_{m_1}| - \mathbf{R}]\|_2\|\mathbf{v}^*\|_2 + \|\mathbf{U}\mathbf{m}^*\|_2)^2} \\
&\leq 1 + \sqrt{1 + \|\mathbf{R}\|_2^2}\sqrt{m_1(\sigma_1 \log_2 m_1)^2 + m_2(\sigma \log_2 m_2)^2} \\
&\qquad + \min(2\sqrt{m_1 m_3}, (\sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3}) \\
&\leq 1 + \sqrt{1 + (\sqrt{m_1} + \sqrt{m_2} + t)^2}\sqrt{m_1(\sigma_1 \log_2 m_1)^2 + m_2(\sigma \log_2 m_2)^2} \\
&\qquad + \min(2\sqrt{m_1}, (\sqrt{m_1} + \sqrt{m_3} + t))\sqrt{m_3} \\
&= \beta_2,
\end{aligned}$$

where the inequalities follow from Equation (3) and Lemma 2.4 except with probability $4e^{-\pi t^2} + 2^{-2\lambda}$. By defining $\mathsf{Abort}_{\{1,2\}} = \mathsf{Abort}_1 \vee \mathsf{Abort}_2$, we obtain

$$\mathbb{P}[\mathbf{w} \text{ valid solution}|\neg\mathsf{Abort}_{\{1,2\}}] \geq 1 - 4e^{-\pi t^2} - 2^{-2\lambda} = 1 - \mathsf{negl}(\lambda). \tag{7}$$

Combining Equations (5), (6) and (7) by the probability chain rule, we get

$$\mathrm{Adv}[\mathcal{B}] \geq (\delta - \mathsf{negl}(\lambda)) \cdot \frac{1}{|\mathcal{T}| - Q} \cdot (1 - \mathsf{negl}(\lambda)) \approx \frac{\delta}{q' - Q},$$

as claimed. □

B.3 Proof of Lemma 3.2

Proof. Consider a PPT adversary $\mathcal{A}$ that can produce a Type II forgery for the signature scheme with advantage δ. We now construct an adversary $\mathcal{B}$ that solves the $\mathrm{SIS}^{\infty,2}_{n,m_1,q,\beta'_\infty,\beta'_2}$ problem. The adversary $\mathcal{B}$ is given $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m_1}$ as input and is asked to find $\mathbf{w} \in \mathcal{L}_q^\perp(\overline{\mathbf{A}})$ such that $0 < \|\mathbf{w}\|_\infty \leq \beta'_\infty$ and $0 < \|\mathbf{w}\|_2 \leq \beta'_2$.

Setup Stage: The adversary $\mathcal{B}$ first generates the tags $\tau^{(1)}, \ldots, \tau^{(Q)}$ that will be used for the signing queries of $\mathcal{A}$ by calling F and incrementing the state st. Note that since F is injective, as mentioned in Remark 3.1, there is no collision among the tags. The adversary $\mathcal{B}$ makes a guess $i^+ \hookleftarrow U([Q])$ on the index of the tag that will be re-used by $\mathcal{A}$ in the forgery stage. Then, $\mathcal{B}$ samples $\mathbf{R} \hookleftarrow U([-1,1]^{m_1 \times m_2})$, and $\mathbf{U}$ from $U([-1,1]^{m_1 \times m_3})$. It then defines

$$\begin{cases} \mathbf{B} = \overline{\mathbf{A}}\mathbf{R} + \tau^{(i^+)}\mathbf{G} \bmod q \\ \mathbf{D} = \overline{\mathbf{A}}\mathbf{U} \bmod q \end{cases}$$

The adversary $\mathcal{B}$ samples $\mathbf{v}$ from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$, $\mathbf{r}_0$ from $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_2}$, and defines

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^+)}}\left(\mathbf{v} - \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{0}_{m_2} \end{bmatrix}\right) \bmod q.$$

Note that for all $i \in [Q]$, we have

$$\mathbf{A}_{\tau^{(i)}} = [\overline{\mathbf{A}} | (\tau^{(i)} - \tau^{(i^+)})\mathbf{G} - \overline{\mathbf{A}}\mathbf{R}],$$

where the contribution in $\mathbf{G}$ vanishes only for $i = i^+$, as there is no collision. The adversary $\mathcal{B}$ thus forms $\mathsf{pp} = (\mathbf{D}; \mathbf{g}; \lambda, n, m_1, m_2, m_3, m, q, \sigma, \sigma_2, \sigma_1)$ and the public key $\mathsf{pk} = (\overline{\mathbf{A}}, \mathbf{B}, \mathbf{u})$, and sends both to $\mathcal{A}$.

Query Stage: We distinguish the queries for $i \neq i^+$ from the i^+-th query. First, consider the i-th query, for $i \neq i^+$, on the message $\mathbf{m}^{(i)}$. $\mathcal{B}$ samples a fresh randomness $\mathbf{r}^{(i)}$ from $\mathcal{D}_{\mathbb{Z}^{m_1}, \sigma_2}$ and computes the commitment $\mathbf{c} = \overline{\mathbf{A}}\mathbf{r}^{(i)} + \mathbf{D}\mathbf{m}^{(i)} \bmod q$. Since $\tau^{(i)} - \tau^{(i^+)} \in \mathbb{Z}_q^\times$, $\mathcal{B}$ computes

$$\mathbf{v}^{(i)} = \mathsf{SampleD}(\mathbf{R}, \mathbf{A}, (\tau^{(i)} - \tau^{(i^+)})\mathbf{I}_n, \mathbf{u} + \mathbf{c}, \sigma) - \begin{bmatrix} \mathbf{r}^{(i)} \\ \mathbf{0}_{m_2} \end{bmatrix}.$$

Note that $\mathbf{v}^{(i)}$ is correctly distributed and passes verification (with overwhelming probability by Lemma 2.2 and 2.3). The signature given to $\mathcal{A}$ is $\mathsf{sig}_i = (\tau^{(i)}, \mathbf{v}^{(i)})$.

Now consider the i^+-th query. In this case, $\mathcal{B}$ simply computes $\mathbf{v}^{(i^+)} = \mathbf{v} - \begin{bmatrix} \mathbf{r}_0 - \mathbf{U}\mathbf{m}^{(i^+)} \\ \mathbf{0}_{m_2} \end{bmatrix}$ and gives $\mathsf{sig}_{i^+} = (\tau^{(i^+)}, \mathbf{v}^{(i^+)})$ to $\mathcal{A}$. We analyze later the distribution of $\mathbf{v}^{(i^+)}$, but notice that the verification equation is verified because of the definition of $\mathbf{u}$.

$$\begin{aligned} \mathbf{A}_{\tau^{(i^+)}}\mathbf{v}^{(i^+)} &= \mathbf{u} + \mathbf{A}_{\tau^{(i^+)}} \begin{bmatrix} \mathbf{U}\mathbf{m}^{(i^+)} \\ \mathbf{0}_{m_2} \end{bmatrix} \bmod q \\ &= \mathbf{u} + \overline{\mathbf{A}}\mathbf{U}\mathbf{m}^{(i^+)} \bmod q \\ &= \mathbf{u} + \mathbf{D}\mathbf{m}^{(i^+)} \bmod q. \end{aligned}$$

Forgery Stage: After at most Q queries, $\mathcal{A}$ outputs a Type II forgery $(\tau^*, \mathbf{v}^*)$ on a new message $\mathbf{m}^*$. If $\mathcal{A}$ fails to output a valid forgery, event that we denote by Abort_1, then $\mathcal{B}$ aborts. We now condition on $\neg\mathsf{Abort}_1$. At this point, $\mathcal{B}$ checks its guess on i^+ and aborts if $\tau^* \neq \tau^{(i^+)}$. We denote this event Abort_2, and further condition on $\neg\mathsf{Abort}_2$. It holds that

$$\mathbf{A}_{\tau^{(i^+)}}\mathbf{v}^{(i^+)} - \mathbf{D}\mathbf{m}^{(i^+)} = \mathbf{u} \bmod q = \mathbf{A}_{\tau^*}\mathbf{v}^* - \mathbf{D}\mathbf{m}^* \bmod q.$$

Since $\mathbf{A}_{\tau^*} = \mathbf{A}_{\tau^{(i^+)}} = \overline{\mathbf{A}}[\mathbf{I}_{m_1} | -\mathbf{R}]$, it holds that

$$\overline{\mathbf{A}}\left([\mathbf{I}_{m_1} | -\mathbf{R}](\mathbf{v}^{(i^+)} - \mathbf{v}^*) - \mathbf{U}(\mathbf{m}^{(i^+)} - \mathbf{m}^*)\right) = \mathbf{0} \bmod q.$$

As a result, $\mathcal{B}$ forms the vector

$$\mathbf{w} = [\mathbf{I}_{m_1} | -\mathbf{R}](\mathbf{v}^{(i^+)} - \mathbf{v}^*) - \mathbf{U}(\mathbf{m}^{(i^+)} - \mathbf{m}^*) \in \mathbb{Z}^{m_1},$$

which is in $\mathcal{L}_q^{\perp}(\overline{\mathbf{A}})$, and returns it as a solution for SIS.

Advantage: We now analyze the advantage of $\mathcal{B}$. We first focus on the distribution of $(\mathsf{pk}, \mathsf{pp})$. Since $m_1 \log_2 3 \geq n \log_2 q + f(\lambda)$, Lemma 2.1 yields

$$\begin{cases} \varDelta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{R} \bmod q), (\overline{\mathbf{A}}, U(\mathbb{Z}_q^{n \times m_2}))) \leq \frac{m_2}{2}\sqrt{\frac{q^n}{3^{m_1}}} \leq m_2 2^{-f(\lambda)/2-1}, \\ \varDelta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{U} \bmod q), (\overline{\mathbf{A}}, U(\mathbb{Z}_q^{n \times m_3}))) \leq \frac{m_3}{2}\sqrt{\frac{q^n}{3^{m_1}}} \leq m_3 2^{-f(\lambda)/2-1}. \end{cases}$$

As $\overline{\mathbf{A}}, \mathbf{R}$ are independent of $\tau^{(i^+)}\mathbf{G}$, it holds that $\varDelta(\mathbf{B}, \overline{\mathbf{A}}\mathbf{R}) \leq m_2 2^{-f(\lambda)/2}$ (by the triangle inequality). Then, let us analyze the distribution of $\mathbf{u}$. Define $\mathbf{A}' = [\overline{\mathbf{A}}| - \overline{\mathbf{A}}\mathbf{R}] \bmod q$. By construction, we have $\mathbf{u} = \mathbf{A}'\mathbf{v}' \bmod q$, where $\mathbf{v}_1'$ is within statistical distance $7\varepsilon/4$ of $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1}$ by Lemma 2.2, and $\mathbf{v}_2'$ is distributed as $\mathcal{D}_{\mathbb{Z}^{m_2},\sigma}$. Fix $f(m) = \omega(\sqrt{\log_2 m})$ such that $\sigma, \sigma_1 \geq f(m)$. Lemma B.3 thus yields that $\mathbf{u}$ is within negligible statistical distance of $U(\mathbb{Z}_q^n)$, conditioning on $\mathbf{A}'$ being uniform. Changing $\mathbf{A}'$ back to $[\overline{\mathbf{A}}| - \overline{\mathbf{A}}\mathbf{R}]$ gives

$$\varDelta(\mathbf{u}, U(\mathbb{Z}_q^n)) \leq \mathsf{negl}(\lambda) + m_2 2^{-f(\lambda)/2-1} = \mathsf{negl}(\lambda).$$

As a result, $(\mathsf{pk}, \mathsf{pp})$ is correctly distributed up to a negligible statistical distance. We now analyze the distribution of the signature that are produced by $\mathcal{B}$. For the i-th query with $i \neq i^+$, the signature is distributed exactly as in the legitimate algorithm. At the i^+-th signing query, the vector $\mathbf{v}_1^{(i^+)}$ is within statistical distance $7\varepsilon/4$ of $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1,\mathbf{z}^+}$, where $\mathbf{z}^+ = \mathbf{U}\mathbf{m}^{(i^+)}$. As before, by Equation (3) obtained by combining Lemma 2.4 and 2.5, we have

$$\left\|\mathbf{z}^+\right\|_2 = \left\|\mathbf{U}\mathbf{m}^{(i^+)}\right\|_2 \leq \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3},$$

except with probability $2e^{-\pi t^2} + 2^{-2\lambda}$. We now measure the closeness of $\mathbf{v}^{(i^+)}$ to the real distribution by using the Rényi divergence of order α for a free parameter $\alpha > 1$. By Lemma B.2 it holds that

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1} \| \mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1,\mathbf{z}^+}) \leq \exp\left(\frac{\alpha\pi}{\sigma_1^2}\left\|\mathbf{z}^+\right\|_2^2\right) \leq e^{\alpha\pi},$$

as $\sigma_1 \geq \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3}$. Combining the probabilities for the distribution of the keys and the signatures, and by the probability preservation properties of the statistical distance and Rényi divergence of Lemma B.1, we have

$$\mathbb{P}[\neg\mathsf{Abort}_1] \geq e^{-\alpha\pi}(\delta - \mathsf{negl}(\lambda))^{\alpha/(\alpha-1)} \geq e^{-\alpha\pi}\delta^{\alpha/(\alpha-1)} - \mathsf{negl}(\lambda). \tag{8}$$

We then optimize over α. The maximum value of the right-hand side is attained for $\alpha^* = 1 + \sqrt{\log_2(1/\delta)/(\pi \log_2 e)}$. Further, since the guess i^+ is independent of $\mathcal{A}$'s view it holds that

$$\mathbb{P}[\neg\mathsf{Abort}_2|\neg\mathsf{Abort}_1] = \frac{1}{Q}. \tag{9}$$

We now analyze the solution constructed by $\mathcal{B}$. We have to show it is non-zero and has norm at most β'. We first focus on the former. We essentially show that for $\mathcal{A}$ to ensure $\mathbf{w} = \mathbf{0}$, it must predict at least one column $\mathbf{u}'$ of $\mathbf{U}$ as $\mathbf{m}^* \neq \mathbf{m}^{(i^+)}$. So we have

$$\begin{aligned}
&\mathbb{P}[\mathbf{w} = \mathbf{0}] \\
&\leq \mathbb{P}_{\mathbf{u}'}[\mathbf{u}' = \mathbf{u}^* : \mathbf{u}^* \leftarrow \mathcal{A}(\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{u}' \bmod q, \mathbf{v}_1^{(i^+)})] \\
&\leq \sqrt{\mathbb{P}_{\mathbf{u}'}[\mathbf{u}' = \mathbf{u}^* : \mathbf{u}^* \leftarrow \mathcal{A}(\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{u}' \bmod q)] \cdot \mathrm{RD}_2(\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1,\mathbf{Um}^{(i^+)}} \| \mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1})} \\
&\quad + 7\varepsilon/4 \\
&\leq \frac{1+\varepsilon}{1-\varepsilon} e^{\pi} \sqrt{\mathbb{P}_{\mathbf{U}}[\mathbf{u}' = \mathbf{u}^* : \mathbf{u}^* \leftarrow \mathcal{A}(\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{u}' \bmod q)]} + 7\varepsilon/4
\end{aligned}$$

where the last inequality stems from Lemma B.2 as $\sigma_1 \geq \sigma \geq \eta_\varepsilon(\mathbb{Z}^{m_1})$[9]. Then, since $m_1 \log_2 3 \geq n \log_2 q + f(\lambda)$ and that $\mathbf{Au}' \bmod q$ can take $2^{n \log_2 q}$ values, [DORS08, Lem. 2.2] then gives that $\mathbf{u}'$ given $\mathbf{Au}' \bmod q$ contains at least $m_1 \log_2 3 - n \log_2 q \geq f(\lambda) = \omega(\log_2 \lambda)$ bits of entropy. Hence, $\mathbf{w} \neq \mathbf{0}$ except with negligible probability.

Finally, it holds that

$$\begin{aligned}
\|\mathbf{w}\|_\infty &\leq \|(\mathbf{v}_1 - \mathbf{r}_0) - \mathbf{v}_1^*\|_\infty + m_2 \|\mathbf{R}\|_{\max} \|\mathbf{v}_2 - \mathbf{v}_2^*\|_\infty + m_3 \|\mathbf{U}\|_{\max} \|\mathbf{m}^*\|_\infty \\
&\leq 2\sigma_1 \log_2 m_1 + m_2 \cdot 2\sigma \log_2 m_2 + m_3 \\
&= \beta'_\infty.
\end{aligned}$$

The inequality is valid if $\mathbf{v}_1 - \mathbf{r}_0$ follows $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma_1}$ (Lemma 2.2) and that the Gaussian tail bound of Lemma 2.3 is verified for $\mathbf{v}_1 - \mathbf{r}_0, \mathbf{v}_2$. By the union bound, this happens with probability at least $1 - (2m_1 e^{-\pi \log_2^2 m_1} + 7\varepsilon/4) - 2m_2 e^{-\pi \log_2^2 m_2} = 1 - \mathsf{negl}(\lambda)$. As in the proof of Lemma 3.1, we cannot use the Gaussian tail bound in ℓ_2 norm for $\mathbf{v}^*$. Hence, we have the following

$$\begin{aligned}
\|\mathbf{w}\|_2 &\leq \sqrt{1 + \|\mathbf{R}\|_2^2} \sqrt{(\sigma^2 m_1 + \sigma_2^2 m_1 + m_1 \sigma_1^2 \log_2^2 m_1) + (\sigma^2 m_2 + m_2 \sigma^2 \log_2^2 m_2)} \\
&\qquad + \|\mathbf{Um}^*\|_2 \\
&\leq \sqrt{1 + (\sqrt{m_1} + \sqrt{m_2} + t)^2} \sqrt{\sigma_1^2 m_1 (1 + \log_2^2 m_1) + \sigma^2 m_2 (1 + \log_2^2 m_2)} \\
&\qquad + \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_3} + t)\sqrt{m_3} \\
&= \beta'_2,
\end{aligned}$$

where the first inequality follows from Lemma 2.3 except with probability $2 \cdot 2^{-2m_1} + 2^{-2m_2}$, and the second inequality stems from Lemma 2.4 except with probability $4e^{-\pi t^2} + 2^{-2\lambda}$. This yields

$$\mathbb{P}[\mathbf{w} \text{ valid solution} | \neg \mathsf{Abort}_1 \wedge \neg \mathsf{Abort}_2] = 1 - \mathsf{negl}(\lambda). \tag{10}$$

[9] Note that the Rényi divergence is taken in the opposite direction than before, hence the presence of the factor $(1+\varepsilon)/(1-\varepsilon)$.

Combining Equations (8), (9) and (10) by the probability chain rule, we get

$$\mathrm{Adv}[\mathcal{B}] \geq (\delta^{\alpha^*/(\alpha^*-1)} e^{-\alpha^* \pi} - \mathsf{negl}(\lambda)) \cdot \frac{1}{Q} \cdot (1 - \mathsf{negl}(\lambda)),$$

as desired. Note that the parameters and the behavior of $\mathcal{B}$ do not depend on the order α that is used to compute the advantage bound. As such, α^* can indeed depend on the forger's advantage δ. □

C Anonymous Credentials

C.1 Definitions

In Section 5.3, we use the definition and security model from [FHS19] that we recall below. We note that, in their definition, the attributes are revealed to the signer during issuance (but not the user's secret key). This means that when using the OblSign protocol of our SEP, part of the message to be signed is revealed to the signer. Rather than an artifact specific to the construction from [FHS19], this peculiarity stems from the difficulty of formally defining a notion of unforgeability when the signed message is hidden, as discussed in Section 5. Regardless, this is not problematic in practice as credentials would usually be emitted on known attributes, even though the latter would be hidden when presenting the credential. Although we prove the security of our construction in this setting, our protocol could easily hide the attributes during issuance if necessary.

C.1.1 Syntax. An anonymous credentials is composed of two algorithms and two interactive protocols. The OKeyGen takes as input public parameters (in our case those generated by Setup) and outputs the organization's key pair $(\mathsf{opk}, \mathsf{osk})$. Similarly, the UKeyGen takes the public parameters (and possibly the organization's public key) and outputs the user's key pair $(\mathsf{upk}, \mathsf{usk})$. The $\mathsf{Issue}_{O,U}$ protocol involves an organization O with $(\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st})$ and the users' attributes $(\mathbf{m}_i)_{i \in [\ell]}$, and a user U with $(\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp})$ and its attributes $(\mathbf{m}_i)_{i \in [\ell]}$. The user either obtains a credential cred on its attributes or $\perp$ if the protocol failed, while the organization just gets notified of whether or not the execution was successful. Finally, the $\mathsf{Show}_{U,V}$ protocol involves a user U with $(\mathsf{usk}, \mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i)_{i \in [\ell]}, \mathsf{cred}, \mathcal{I})$ and a verifier V with $(\mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i)_{i \in \mathcal{I}})$. The protocol outputs $b = 1$ to V if the credential cred is valid for the disclosed attributes $(\mathbf{m}_i)_{i \in \mathcal{I}}$ and $b = 0$ otherwise, and U gets no output.

C.1.2 Security Requirements. The model from [FHS19] stipulates that the anonymous credentials system must be *correct*, *anonymous*, and *unforgeable*. Correctness says that an honest execution of $\mathsf{Issue}_{O,U}$ should succeed often[10], and that an honest execution of $\mathsf{Show}_{U,V}$ on honestly obtained credentials should

[10] There may be a correctness error which is similar to the completeness error of zero-knowledge arguments.

also succeed often. For the anonymity and unforgeability, we stick to the syntax of [FHS19] by first defining the following variables and oracles.

- HU: Set of user indices corresponding to honest users, initially empty.
- CU: Set of user indices corresponding to corrupt users, initially empty.
- ctr: Issuance counter, initially set to 0.
- A: Set of triplets $(j, j', (\mathbf{m}_i)_{i\in[\ell]})$ filled when the oracles $\mathcal{O}_{\mathsf{ObtIss}}$ or $\mathcal{O}_{\mathsf{Issue}}$ successfully issue a credential for user j on the attributes $(\mathbf{m}_i)_{i\in[\ell]}$, and where j' is an issuance index.
- $\mathcal{O}_{\mathsf{HU}}(j)$: Given a user index j, it returns $\perp$ if $j \in \mathsf{HU} \cup \mathsf{CU}$. Otherwise, it samples $(\mathsf{upk}_j, \mathsf{usk}_j) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$ and returns upk_j. It then adds j to HU.
- $\mathcal{O}_{\mathsf{CU}}(j, \mathsf{upk})$: Given a user index j and optionally a public key upk, it registers a new user with public key upk if $j \notin \mathsf{HU}$. Otherwise, it returns usk_j and sets $\mathsf{HU} \leftarrow \mathsf{HU} \setminus \{j\}$. Either way, it adds j to CU. The former case models the ability to register users with malformed keys, i.e., who do not know the associated secret key.
- $\mathcal{O}_{\mathsf{ObtIss}}(j, \mathbf{m})$: On input $j \in \mathsf{HU}$ and attributes $\mathbf{m} = (\mathbf{m}_i)_{i\in[\ell]}$, the oracle runs $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}_j, \mathsf{pp}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}_j, \mathsf{upk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$ assuming the roles of both O and user j. If the execution succeeds, it increments the issuance counter ctr, it stores the obtained credential and stores $(j, \mathsf{ctr}, \mathbf{m})$ in A. It returns indication of whether the execution succeeded. If $j \notin \mathsf{HU}$, it simply returns $\perp$.
- $\mathcal{O}_{\mathsf{Obtain}}(j, \mathbf{m})$: Given a user index j and attributes $\mathbf{m}$, it returns $\perp$ if $j \notin \mathsf{HU}$. Otherwise, it runs $\mathsf{Issue}_{\mathcal{A},U}(\cdot, (\mathsf{usk}_j, \mathsf{upk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$ with the adversary $\mathcal{A}$ posing as the organization. It thus issues a credential to an honest user.
- $\mathcal{O}_{\mathsf{Issue}}(j, \mathbf{m})$: Given a user index j and attributes $\mathbf{m}$, it returns $\perp$ if $j \notin \mathsf{CU}$. Otherwise, it runs $\mathsf{Issue}_{O,\mathcal{A}}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}_j, \mathsf{pp}, \mathsf{st}, \mathbf{m}), \cdot)$ with the adversary assuming the role of the user. It thus allows the adversary to obtain credentials from the honest organization. If the execution is successful, it increments the issuance counter ctr, stores the credential and adds $(j, \mathsf{ctr}, \mathbf{m})$ in A.
- $\mathcal{O}_{\mathsf{Show}}(j', (\mathbf{m}_i^{(j')})_{i\in\mathcal{I}})$: It takes as input an issuance index j' and disclosed attributes $(\mathbf{m}_i^{(j')})_{i\in\mathcal{I}}$. The issuance index corresponds to a successfully issued credential $\mathsf{cred}^{(j')}$ on $(\mathbf{m}_i^{(j')})_{i\in[\ell]}$ for a user j during the j'-th query to $\mathcal{O}_{\mathsf{IssObt}}$ or $\mathcal{O}_{\mathsf{Obtain}}$. If the corresponding user j is not honest, i.e., $j \notin \mathsf{HU}$, the oracle returns $\perp$. Else, it runs $\mathsf{Show}_{U,\mathcal{A}}((\mathsf{usk}_j, \mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i^{(j')})_{i\in[\ell]}, \mathsf{cred}^{(j')}, \mathcal{I}), \cdot)$ with the adversary posing as the verifier.

Anonymity. The anonymity property captures the fact that a user showing its credential cred obtained on attribute vector $\mathbf{m}$ remains anonymous among all users who have the same disclosed attributes $(\mathbf{m}_i)_{i\in\mathcal{I}}$. It means that during an execution of $\mathsf{Show}_{U,V}$, the verifier V does not learn anything other that U owns a valid credential on the shown attributes $(\mathbf{m}_i)_{i\in\mathcal{I}}$. Additionally, different showings of the same credential with the same revealed attributes should be unlinkable. It is formalized by the game presented in Figure C.1. The anonymous credentials system is *anonymous* if for all PPT adversary $\mathcal{A}$, its advantage $|\mathbb{P}[b^* = b \wedge \mathcal{O}_{\mathsf{CU}}$ was not queried on j_0 nor $j_1] - 1/2|$ in the anonymity game is negligible. To avoid overloading the protocols we assume that $(\mathsf{opk}, \mathsf{osk})$ are honestly generated, but this assumption is not necessary if one includes a zero-knowledge proof that they know the secret osk linked to opk.

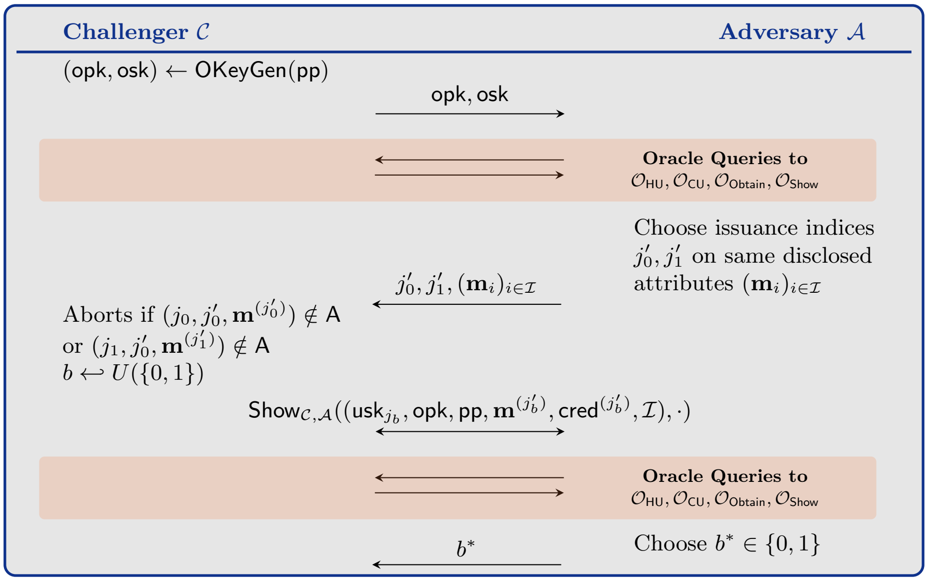
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$
$\mathsf{opk}, \mathsf{osk}$
Oracle Queries to
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Show}}$
Choose issuance indices j'_0, j'_1 on same disclosed attributes $(\mathbf{m}_i)_{i \in \mathcal{I}}$
$j'_0, j'_1, (\mathbf{m}_i)_{i \in \mathcal{I}}$
Aborts if $(j_0, j'_0, \mathbf{m}^{(j'_0)}) \notin \mathsf{A}$
or $(j_1, j'_0, \mathbf{m}^{(j'_1)}) \notin \mathsf{A}$
$b \hookleftarrow U(\{0,1\})$
$\mathsf{Show}_{\mathcal{C},\mathcal{A}}((\mathsf{usk}_{j_b}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}^{(j'_b)}, \mathsf{cred}^{(j'_b)}, \mathcal{I}), \cdot)$
Oracle Queries to
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Show}}$
b^*
Choose $b^* \in \{0,1\}$

Fig. C.1. Anonymity Game for the Anonymous Credentials System. The index j_α is the user index associated to the issuance index j'_α. The attribute vector $\mathbf{m}^{(j'_\alpha)}$ is the attribute vector used in the j'_α issuance, and must satisfy $(\mathbf{m}_i^{(j'_\alpha)})_{i \in \mathcal{I}} = (\mathbf{m}_i)_{i \in \mathcal{I}}$.

Unforgeability. The unforgeability property of anonymous credentials ensures that a user cannot show attributes for which it does not own a valid credential. It means that it cannot impersonate an honest user (as it would mean knowing its secret key) which thwarts replay attacks, and that it cannot forge fresh credentials that have not been issued by the Issue protocol. Additionally, malicious users cannot collude and use their legitimate credentials to obtain a new one on a set of attributes that has not been used in a successful issuance. We formalize it as a game in Figure C.2. The adversary wins the game if the challenger does not abort and if the challenger's output of the execution of Show is $b = 1$. We say that the anonymous credentials system is *unforgeable* if for all PPT adversary $\mathcal{A}$, its probability of winning is negligible.

C.1.3 Additional Assumptions. In order to prove the unforgeability of our anonymous credentials system, we require the inhomogeneous variant of M-SIS, and M-LWE (in knapsack form) which we now recall. Note that the knapsack form of M-LWE is at least as hard as the standard definition by the duality result of [BJRW23, Lem. 4.1].

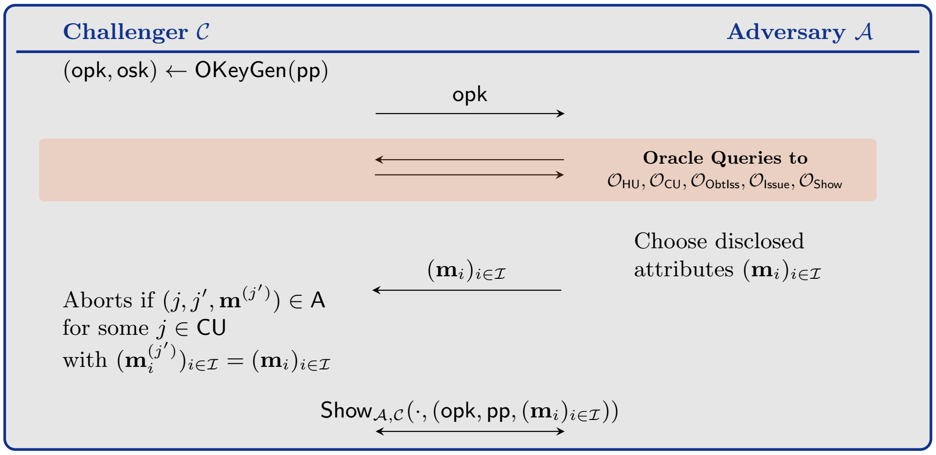
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$
opk
Oracle Queries to
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{ObtIss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}$
Choose disclosed
attributes $(\mathbf{m}_i)_{i\in\mathcal{I}}$
$(\mathbf{m}_i)_{i\in\mathcal{I}}$
Aborts if $(j, j', \mathbf{m}^{(j')}) \in \mathsf{A}$
for some $j \in \mathsf{CU}$
with $(\mathbf{m}_i^{(j')})_{i\in\mathcal{I}} = (\mathbf{m}_i)_{i\in\mathcal{I}}$
$\mathsf{Show}_{\mathcal{A},\mathcal{C}}(\cdot, (\mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i)_{i\in\mathcal{I}}))$

Fig. C.2. Unforgeability Game for the Anonymous Credentials System.

Definition C.1 (M-ISIS)**.** *Let n be a power of two and R the $2n$-th cyclotomic ring. Let d, m, q be positive integers and $\beta > 0$. The* Module Inhomogeneous Short Integer Solution *problem* M-ISIS$_{d,m,q,\beta}$ *asks to find* $\mathbf{s} \in R^m$ *such that* $\mathbf{D}\mathbf{s} = \mathbf{t} \bmod qR$ *and* $\|\mathbf{s}\|_2 \leq \beta$*, given* $\mathbf{D} \hookleftarrow U(R_q^{d\times m})$ *and* $\mathbf{t} \hookleftarrow U(R_q^d)$*.*

Definition C.2 (M-LWE)**.** *Let n be a power of two and R the $2n$-th cyclotomic ring. Let d, m, q be positive integers and $\beta > 0$. The* Module Learning With Errors *problem* M-LWE$_{d,m,q,U(S_{bin})}$ *asks to distinguish between the following distributions: (1)* $(\mathbf{D}, \mathbf{D}\mathbf{s} \bmod qR)$*, where* $\mathbf{D} \sim U(R_q^{d\times m})$ *and* $\mathbf{s} \sim U(S_{bin}^m)$*, and (2)* $(\mathbf{D}, \mathbf{t})$*, where* $\mathbf{D} \sim U(R_q^{d\times m})$ *and* $\mathbf{t} \sim U(R_q^d)$*.*

C.2 Security Analysis

C.2.1 Correctness. We first show correctness of our anonymous credentials, meaning that honest execution of the issuance protocols does not fail, and that honestly obtained credentials can be shown successfully.

Lemma C.1. *The anonymous credentials system of Section 5.3 is correct.*

Proof. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$. Let $(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$ and $(\mathsf{upk}, \mathsf{usk}) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$. Then, let $\mathbf{m} \in S_{bin}^{m_3'}$ and $\mathcal{I} \subseteq [\ell]$. We now consider an honest execution of $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$. By the completeness of the zero-knowledge argument of knowledge, we only have to check the abort condition of step 11. First, note that $\tau \in \mathcal{T}_w$ and $\widetilde{\mathbf{m}} \in S_{bin}^{m_3}$. Then, we have

$$\begin{aligned}\mathbf{A}_\tau \mathbf{v} &= \mathbf{A}_\tau \mathbf{v}' - \mathbf{A}\mathbf{r}' \\ &= \mathbf{u} + \mathbf{c}' - \mathbf{A}(\mathbf{r}'' + \mathbf{r}') \bmod qR \\ &= \mathbf{u} + \mathbf{c} - \mathbf{A}\mathbf{r}' \bmod qR \\ &= \mathbf{u} + \mathbf{D}_s \mathbf{s} + \sum_{i \in [\ell]} \mathbf{D}_i \mathbf{m}_i \bmod qR \\ &= \mathbf{u} + \mathbf{D}\widetilde{\mathbf{m}} \bmod qR.\end{aligned}$$

Finally, it holds that $\mathbf{v}$ is statistically close to an elliptical discrete Gaussian distribution, where $\mathbf{v}_1$ is of width $\sqrt{\sigma^2 + \sigma_3^2 + \sigma_4^2} = \sigma_1$, and $\mathbf{v}_2$ is of width σ. Thence, $\|\mathbf{v}\|_2 \leq \sqrt{\sigma_1^2 n m_1 + \sigma^2 n m_2}$, except with probability $2^{-2(m_1+m_2)}$. This shows that an honest execution of Issue succeeds with overwhelming probability conditioned on the zero-knowledge argument to pass.

We now consider a successful execution of the credential issuance process, i.e., $(\perp; \mathsf{cred}) \leftarrow \mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$. Because the execution did not abort, it means that $\mathsf{Verify}(\mathsf{pk}, \widetilde{\mathbf{m}}, (\tau, \mathbf{v}), \mathsf{pp}) = 1$. Directly from the completeness of the zero-knowledge argument of knowledge, we get that $\mathsf{Show}_{U,V}((\mathsf{usk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}, (\tau, \mathbf{v}), \mathcal{I}); (\mathsf{opk}, \mathsf{pp}, (\mathbf{m}_i)_{i \in \mathcal{I}}))$ outputs $(\perp, 1)$. □

C.2.2 Anonymity. We now prove that the showing of credentials to verifiers or organizations does not leak information about the user's concealed attributes and credential except that they own a valid credential on the disclosed attributes.

Lemma C.2. *The anonymous credentials of Section 5.3 is anonymous based on the zero-knowledge property of the ZKAoK involved in* Show*.*

Proof. We proceed by a single game hop. We define a modified version of the anonymity game of Figure C.1. It is exactly the same game except that when interacting with the adversary in $\mathsf{Show}_{\mathcal{C},\mathcal{A}}((\mathsf{usk}_{j_b}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}^{(j'_b)}, \mathsf{cred}^{(j'_b)}, \mathcal{I}), \cdot)$, the challenger $\mathcal{C}$ will simulate the interaction, i.e., without resorting to usk_{j_b}, $(\mathbf{m}_i^{(j'_b)})_{i \notin \mathcal{I}}$, $\mathsf{cred}^{(j'_b)}$. By the zero-knowledge property of the zero-knowledge argument, the advantage of $\mathcal{A}$ in the modified game is negligibly close to that it would have in the original game.

Now, the view of $\mathcal{A}$ only depends on $(\mathbf{m}_i^{(j'_b)})_{i \in \mathcal{I}}$, which does not depend on b as we require $(\mathbf{m}_i^{(j'_0)})_{i \in \mathcal{I}} = (\mathbf{m}_i)_{i \in \mathcal{I}} = (\mathbf{m}_i^{(j'_1)})_{i \in \mathcal{I}}$. Thence, the view of $\mathcal{A}$ is independent of b and therefore its advantage is 0. It proves that the advantage of $\mathcal{A}$ in the original anonymity game is negligible. □

C.2.3 Unforgeability. We finally prove that a user cannot prove knowledge of a credential they did not receive from a successful execution of the issuance protocol. At a high level, if an adversary is able to do so, then it either breaks the soundness of the ZKAoK or was able to forge a signature.

Lemma C.3. *The anonymous credentials of Section 5.3 is unforgeable based on the hardness of* M-ISIS$_{d,m_s,q,\sqrt{nm_s}}$*, the zero-knowledge property of the ZKAoK involved in* Issue *and* Show*, on the soundness of the ZKAoK involved in* Show*, and on the EUF-CMA security of the signature scheme of Section 3.*

Proof. We consider a PPT adversary $\mathcal{A}$ against the unforgeability game. As described, $\mathcal{A}$ receives the organization's public key opk and must return a disclosed set of attributes $(\mathbf{m}_i^*)_{i\in\mathcal{I}}$ while proving possession of a credential cred^* on said attributes in a successful execution of Show with the honest organization. If $(\mathbf{m}_i^*)_{i\in\mathcal{I}}$ corresponds to an attribute vector $\mathbf{m}$ that was queried for issuance by a corrupt user, the forgery is refused. This leaves two scenarios. Either (1) $\mathcal{A}$ tried to impersonate an honest user, or (2) it did not. Since $\mathcal{A}$ must prove knowledge of a secret $\mathbf{s}^*$ satisfying $\mathbf{D}_s\mathbf{s}^* = \mathbf{t}$, this means that (1) corresponds to the case where there exists $j \in \mathsf{HU}$ such that $\mathbf{s}^* = \mathsf{usk}_j$, i.e., verifying $\mathbf{D}_s\mathbf{s}^* = \mathsf{upk}_j \bmod qR$, and (2) where for every $j \in \mathsf{HU}$, $\mathbf{s}^* \neq \mathsf{usk}_j$. We tackle these two types of forgeries separately.

<u>(1) Impersonation Forgery.</u> At the outset, the challenger receives an M-ISIS instance $(\overline{\mathbf{D}_s}, \overline{\mathbf{t}})$. It then runs Setup but sets $\mathbf{D}_s = \overline{\mathbf{D}_s}$ instead of sampling it themselves. It then makes a guess on which honest user will be targeted. For that it samples $j^+ \hookleftarrow U([|\mathcal{T}_w|])$. Indeed, the number of users requesting credentials to one organization is bounded by the number of possible tags as explained in Remark 3.1. It then runs $\mathsf{OKeyGen}(\mathsf{pp})$ to obtain $(\mathsf{opk}, \mathsf{osk}) = ((\mathbf{A}, \mathbf{B}, \mathbf{u}), \mathbf{R})$, and sends opk to $\mathcal{A}$. From there on out the adversary makes queries to different oracles which are answered as follows.

- $\mathcal{O}_{\mathsf{HU}}$: On input a user index j, the challenger runs $(\mathsf{upk}_j, \mathsf{usk}_j) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$ and outputs upk_j if $j \neq j^+$, and outputs $\overline{\mathbf{t}}$ if $j = j^+$.
- $\mathcal{O}_{\mathsf{CU}}$: On input a user index j, it gives usk_j to $\mathcal{A}$ if $j \neq j^+$. If $j = j^+$, the challenger aborts the reduction altogether as the guess was wrong.
- $\mathcal{O}_{\mathsf{ObtIss}}$: It takes as input a user index j and attribute vector $\mathbf{m} \in S_{bin}^{m_3'}$. If $j \in \mathsf{CU}$ it sends $\perp$ to $\mathcal{A}$. Otherwise, if $j \neq j^+$, the challenger can assume the role of the issuer and the user in the Issue protocol as it knows the issuer's key osk and the key usk_j of user j. If the execution fails, it sends $\perp$ to $\mathcal{A}$, and nothing if it succeeds. However, if $j = j^+$, it instead generates $\mathbf{c}$ as $\mathbf{A}\mathbf{r}' + \overline{\mathbf{t}} + \sum_i \mathbf{D}_i\mathbf{m}_i \bmod qR$, and simulates the zero-knowledge argument when assuming the role of the user in Step 4 of Issue. Again, if this modified execution fails, it sends $\perp$ to $\mathcal{A}$, and nothing if it succeeds.
- $\mathcal{O}_{\mathsf{Issue}}$: It takes as input a user index j and attribute vector $\mathbf{m} \in S_{bin}^{m_3'}$. If $j \notin \mathsf{CU}$, it returns $\perp$ to $\mathcal{A}$ and does not engage in the issuance protocol. Otherwise, since the challenger knows osk, it can run the Issue protocol where the adversary plays the user j with public key upk_j, and the challenger plays the signer. In the end, either $\mathcal{A}$ gets $\perp$ if the execution failed, or obtained a credential $(\tau, \mathbf{v})$ on $\mathbf{m}$.
- $\mathcal{O}_{\mathsf{Show}}$: It takes an issuance index j', corresponding to the j'-th credential issued on $(\mathbf{m}_i^{(j')})_{i\in[\ell]}$ for some user j, and also disclosed attributes $(\mathbf{m}_i^{(j')})_{i\in\mathcal{I}}$.

If user $j \in \mathsf{CU}$, the challenger outputs $\perp$ to $\mathcal{A}$. Otherwise, if $j \neq j^+$, it runs the legitimate protocol Show where $\mathcal{A}$ assumes the role of the verifier, which can be done as the challenger knows usk_j, the attributes and the credential. If $j = j^+$ however, it cannot run Show. Instead, it simulates the zero-knowledge argument with the adversary as the verifier.

If the guess j^+ is correct, which implies that j^+ is never queried to $\mathcal{O}_{\mathsf{CU}}$, then the game is correctly simulated. Indeed, the differences stem from the public key of user j^+ and the simulation of the zero-knowledge arguments. Since $\overline{\mathbf{t}}$ is uniform, it is indistinguishable from regular keys $\mathbf{D}_s\mathbf{s}$ under M-LWE$_{d,2d,q,U(S_{bin})}$. Then, by the zero-knowledge property of the ZKAoK, the simulated proof are correctly distributed. Hence, if $\mathcal{A}$ has advantage δ in performing a forgery attack satisfying (1), it can successfully prove knowledge of $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}^*}, \tau^*, \mathbf{v}^*)$ with disclosed attributes $(\mathbf{m}_i^*)_{i \in \mathcal{I}^*}$ such that $\mathsf{Verify}(\mathsf{opk}, \widetilde{\mathbf{m}}^*, (\tau^*, \mathbf{v}^*), \mathsf{pp}) = 1$ where $\widetilde{\mathbf{m}}^* = [\mathbf{s}^{*T} | \mathbf{m}_1^{*T} | \ldots | \mathbf{m}_k^{*T}]^T$. The challenger can then extract $\mathbf{s}^*$ by the soundness of the ZKAoK, and because it is an attack of type (1), there exists $j^* \in \mathsf{HU}$ such that $\mathbf{s}^* = \mathsf{usk}_j$, thus implying $\mathbf{D}_s\mathbf{s}^* = \mathsf{upk}_{j^*}$. If $j^* = j^+$, the challenger's guess is correct and this happens with probability at least $1/|\mathcal{T}_w|$ because it means j^+ was never queried to $\mathcal{O}_{\mathsf{CU}}$ and was therefore independent of the view of $\mathcal{A}$. In that case, we thus have $\overline{\mathbf{D}_s}\mathbf{s}^* = \overline{\mathbf{t}} \bmod qR$, and $\mathbf{s}^* \in S_{bin}^{m_s}$ so $\|\mathbf{s}^*\|_2 \leq \sqrt{nm_s}$. The challenger thus solves the M-ISIS instance with advantage at least $\delta/|\mathcal{T}_w| - \mathsf{negl}(\lambda)$.

(2) Fresh Forgery. If the challenger expects this type of forgery, it expects a forgery on the SEP. It therefore flips a coin to guess which type of forgery on the signature will be performed (type I or type II).

If it expects a type I forgery, it proceeds exactly as in Section B.2, without having to extract the commitment randomness in the issuance. This is because signature queries are answered legitimately without having to tamper with the randomness. As a result, once the challenger has changed the setup, it can answer all the oracle queries $\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{ObtIss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}$ legitimately. When $\mathcal{A}$ eventually proves knowledge of $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}}, \tau^*, \mathbf{v}^*)$ with disclosed attributes $(\mathbf{m}_i^*)_{i \in \mathcal{I}}$ such that $\mathsf{Verify}(\mathsf{opk}, \widetilde{\mathbf{m}}^*, (\tau^*, \mathbf{v}^*), \mathsf{pp}) = 1$, the challenger can extract $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}}, \tau^*, \mathbf{v}^*)$ from the proof by the soundness of the ZKAoK. Then, $(\tau^*, \mathbf{v}^*)$ is a valid type I forgery for our SEP as $\widetilde{\mathbf{m}}^*$ is a fresh message. Indeed, by definition of type (2) forgeries of the anonymous credentials, we have that $\mathbf{s}^* \neq \mathsf{usk}_j$ for all $j \in \mathsf{HU}$. This first fact means that $\widetilde{\mathbf{m}}^*$ differs from all the $\widetilde{\mathbf{m}}$ involved in calls to $\mathcal{O}_{\mathsf{ObtIss}}$. Secondly, by the definition of a forgery of the AC, it must hold that for all $j \in \mathsf{CU}$, $(j, j', (\mathbf{m}_i^*)_{i \in [\ell]}) \notin \mathsf{A})$, which means that $\widetilde{\mathbf{m}}^*$ must differ from all the $\widetilde{\mathbf{m}}$ involved in calls to $\mathcal{O}_{\mathsf{Issue}}$. As a result, we obtain a solution to M-SIS as in Section B.2.

If it expects a type II forgery of the SEP, it proceeds as in Section B.3 with a minor difference due to the fact that it needs to control the commitment randomness for the i^+-th signature query, as explained in Remark 5.1. In this context, in the issuance corresponding to the tag $\tau^* = \tau^{(i^+)}$ that will be used in the forgery extracted from the showing of the AC, the challenger proceeds as follows. By the soundness of the ZKAoK, it extracts $(\mathbf{r}'^{(i^+)}, \mathbf{s}^{(i^+)})$ such that $\mathbf{c}^{(i^+)} =$

$\mathbf{A}\mathbf{r}'^{(i^+)} + \mathbf{D}_s\mathbf{s}^{(i^+)} + \sum_{i\in[\ell]} \mathbf{D}_i\mathbf{m}_i^{(i^+)} \bmod qR$. As opposed to Section B.3 where it computed $\mathbf{v}'^{(i^+)} = \mathbf{v} - [(\mathbf{r}_0 - \mathbf{U}\widetilde{\mathbf{m}}^{(i^+)})^T|\mathbf{0}]^T$, with $\widetilde{\mathbf{m}}^{(i^+)} = [{\mathbf{s}^{(i^+)}}^T|{\mathbf{m}^{(i^+)}}^T]^T$, here, it computes

$$\mathbf{v}'^{(i^+)} = \mathbf{v} - \begin{bmatrix} \mathbf{r}_0 - \mathbf{U}\widetilde{\mathbf{m}}^{(i^+)} - \mathbf{r}'^{(i^+)} \\ \mathbf{0} \end{bmatrix}.$$

The rest of the proof remains the same. In the end, when $\mathcal{A}$ engages in Show to attack the unforgeability of the AC, the challenger extracts $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i\notin\mathcal{I}}, \tau^*, \mathbf{v}^*)$. It thus obtain $(\tau^*, \mathbf{v}^*)$ which is a valid type II forgery for the SEP on message $\widetilde{\mathbf{m}}^*$ (which is fresh as explained above). It then proceeds exactly as in Section B.3 to obtain a solution to M-SIS.

In the end, if $\mathcal{A}$ has advantage δ in producing a forgery of type (2) for the anonymous credentials system, it holds that $\delta \leq \sup_{\mathcal{A}'\ \mathsf{PPT}} \mathrm{Adv}^{EUF-CMA}[\mathcal{A}'] + \mathsf{negl}(\lambda)$, where $\mathrm{Adv}^{EUF-CMA}[\mathcal{A}']$ denotes the advantage of $\mathcal{A}'$ in producing a valid forgery (type I or type II) for our signature with efficient protocols. □

D Proof of Lemma 4.1

Proof. For clarity, we denote by $\overline{\mathbf{x}} = \mathbf{x} \bmod q$ the vector of representatives in $[-q/2, q/2]$ of a vector $\mathbf{x}$. With such representatives, we have $\overline{C} = C$ for any integer C in $[-q/2, q/2]$, which simplifies the notations in what follows. We assume that there exists i in $[n]$ such that $\overline{w_i} \notin [-2B, 2B]$. Let $\mathbf{h}$ be a random vector distributed according to $U([-1,1]^n)$. For clarity, we define $S = [n] \setminus \{i\}$. We now have

$$\begin{aligned}
&\mathbb{P}_{\mathbf{h}}[\overline{\mathbf{h}^T\mathbf{w}} \in [-B,B]] \\
&\quad= \sum_{C\in[-q/2,q/2]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_j w_j} = C] \cdot \mathbb{P}_{\mathbf{h}}[\overline{\sum_{i\in[n]} h_j w_j} \in [-B,B] | \overline{\sum_{j\in S} h_j w_j} = C] \\
&\quad= \sum_{C\in[-q/2,q/2]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_j w_j} = C] \cdot \mathbb{P}_{h_i}[\overline{h_i\overline{w_i} + C} \in [-B,B]] \\
&\quad= \sum_{C\in[-q/2,q/2]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_j w_j} = C] \cdot \frac{\left|\{h_i : \overline{h_i\overline{w_i} + C} \in [-B,B]\}\right|}{3},
\end{aligned}$$

where the second equality is a consequence of h_i being uniform in $[-1,1]$. We now split the sum indexed by C into two complementary parts Σ_1 and Σ_2 as follows.

$$\Sigma_1 = \sum_{C\in(-\frac{q}{2}+2B,\frac{q}{2}-2B)\cap\mathbb{Z}} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_j w_j} = C] \cdot \frac{\left|\{h_i : \overline{h_i\overline{w_i} + C} \in [-B,B]\}\right|}{3}$$

$$\Sigma_2 = \sum_{C\in[-\frac{q}{2},-\frac{q}{2}+2B]\cup[\frac{q}{2}-2B,\frac{q}{2}]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_j w_j} = C] \cdot \frac{\left|\{h_i : \overline{h_i\overline{w_i} + C} \in [-B,B]\}\right|}{3}$$

We can now focus on bounding the set $\{h_i : \overline{h_i\overline{w_i}+C} \in [-B,B]\}$ in each case. First note that for all $h_i \in [-1,1]$, if

$$\begin{cases} \overline{h_i\overline{w_i}+C} \in [-B,B] \\ \overline{(h_i+1)\overline{w_i}+C} \in [-B,B], \end{cases}$$

are both satisfied, then there exist $r_1, r_2 \in [-B,B]$ and $k_1, k_2 \in \mathbb{Z}$, such that

$$h_i\overline{w_i}+C = r_1 + k_1q \quad \wedge \quad (h_i+1)\overline{w_i}+C = r_2+k_2q.$$

Note that the above equations are now over $\mathbb{Z}$, not $\mathbb{Z}_q$. Combining these two equations gives us $\overline{w_i} = r_2 - r_1 + (k_2-k_1)q$, which implies that the representative $\overline{w_i}$ is necessarily in $[-2B,2B]$. This contradicts the original assumption of $\overline{w_i} \notin [-2B,2B]$. In other words, we have shown that the set $\{h_i : \overline{h_i\overline{w_i}+C} \in [-B,B]\}$ cannot contain two consecutive numbers.

Now let us consider the situation where both h_i and h_i+2 would be in this set. As $h_i \in [-1,1]$, this can only occur when $h_i = -1$. This means that:

$$\begin{cases} \overline{-\overline{w_i}+C} \in [-B,B] \\ \overline{\overline{w_i}+C} \in [-B,B], \end{cases}$$

and hence there exist $r_1, r_2 \in [-B,B]$ and $k_1,k_2 \in \mathbb{Z}$, such that

$$-\overline{w_i}+C = r_1+k_1q \tag{11}$$
$$\overline{w_i}+C = r_2+k_2q. \tag{12}$$

This implies that $2\overline{w_i} = r_2 - r_1 + (k_2-k_1)q$. As $\overline{w_i} \notin [-2B,2B]$, these equations can be satisfied only when $\overline{w_i} = \frac{r_2-r_1\pm q}{2}$ with $r_2 - r_1 \in [-2B,2B]$. The latter interval implies that $\overline{w_i} \in [-\frac{q}{2}, -\frac{q}{2}+B] \cup [\frac{q}{2}-B, \frac{q}{2}]$. But in that case, Equation (11) implies that $C = \overline{w_i} + r_1 \in [-\frac{q}{2}, -\frac{q}{2}+2B] \cup [\frac{q}{2}-2B, \frac{q}{2}]$. In other words, $\{h_i : \overline{h_i\overline{w_i}+C} \in [-B,B]\}$ contains at most 1 element if $C \notin [-\frac{q}{2}, -\frac{q}{2}+2B] \cup [\frac{q}{2}-2B, \frac{q}{2}]$ and at most two elements otherwise. We thus get the following bounds on $\varSigma_1$ and $\varSigma_2$.

$$\varSigma_1 \leq \frac{1}{3} \sum_{C \in (-\frac{q}{2}+2B, \frac{q}{2}-2B)\cap\mathbb{Z}} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_jw_j} = C]$$
$$\varSigma_2 \leq \frac{2}{3} \sum_{C \in [-\frac{q}{2}, -\frac{q}{2}+2B] \cup [\frac{q}{2}-2B, \frac{q}{2}]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_jw_j} = C]$$

Let x denote $\sum_{C \in [-\frac{q}{2}, -\frac{q}{2}+2B] \cup [\frac{q}{2}-2B, \frac{q}{2}]} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_jw_j} = C]$. Then, we have that $1-x$ is $\sum_{C \in (-\frac{q}{2}+2B, \frac{q}{2}-2B)\cap\mathbb{Z}} \mathbb{P}_{\mathbf{h}}[\overline{\sum_{j\in S} h_jw_j} = C]$ which yields

$$\mathbb{P}_{\mathbf{h}}[\overline{\mathbf{h}^T\mathbf{w}} \in [-B,B]] \leq \frac{1}{3} + \frac{x}{3}.$$

Our last task is then to find a suitable upper bound on x. More concretely, we want to prove that $x \leq \frac{2}{3}$. To this end, we will show that, for any vector $\mathbf{u}$ uniformly sampled from $\{-1, 0, 1\}^{n-1}$ and any vector $\mathbf{w} \in \mathbb{Z}_q^{n-1}$, the probability (over the choice of $\mathbf{u}$) that $\sum_j u_j w_j \in (-\frac{q}{2}, -\frac{q}{2} + 2B] \cup [\frac{q}{2} - 2B, \frac{q}{2}]$ is at most $\frac{2}{3}$ when the requirements of our lemma are fulfilled.

In our case, we first recall that the elements of the sets $\{h_j\}_{j \in S}$ are uniformly sampled from $\{-1, 0, 1\}^{n-1}$ that we identify to $\mathbb{Z}_3$ seen as an additive group. Let $\mathbf{t} = [1, \ldots, 1]^T \in \mathbb{Z}_3^{n-1}$ and let $T = \mathbb{Z}_3^{n-1} / \langle \mathbf{t} \rangle$. Any element $\mathbf{u} \in T$ then has exactly 3 representatives in $\mathbb{Z}_3^{n-1}$ that we note $\mathbf{u}, \mathbf{u}', \mathbf{u}'' \in \{-1, 0, 1\}^{n-1}$. We then have exactly

$$\mathbf{u} + \mathbf{u}' + \mathbf{u}'' = \mathbf{0}$$

where the previous equation holds in $\mathbb{Z}^{n-1}$ because, for any $j \in [n-1]$, it holds $\{u_j, u'_j, u''_j\} = \{-1, 0, 1\}$. For all $\mathbf{w} \in \mathbb{Z}_q^{n-1}$, we define

$$\Sigma_{\mathbf{u}} = \sum_{j=1}^{n-1} u_j w_j, \;\; \Sigma_{\mathbf{u}'} = \sum_{j=1}^{n-1} u'_j w_j, \;\; \Sigma_{\mathbf{u}''} = \sum_{j=1}^{n-1} u''_j w_j.$$

We then know that $\Sigma_{\mathbf{u}} + \Sigma_{\mathbf{u}'} + \Sigma_{\mathbf{u}''} = 0$ in $\mathbb{Z}$ and hence that $\overline{\Sigma_{\mathbf{u}}} + \overline{\Sigma_{\mathbf{u}'}} + \overline{\Sigma_{\mathbf{u}''}} = 0$. What remains to prove is that this implies that at least one of these representatives is not in $(-\frac{q}{2}, -\frac{q}{2} + 2B] \cup [\frac{q}{2} - 2B, \frac{q}{2}]$. Let us assume, without loss of generality that both $\overline{\Sigma_{\mathbf{u}}}$ and $\overline{\Sigma_{\mathbf{u}'}}$ are in this set (else, we are done). This means (over $\mathbb{Z}$) that

$$\overline{\Sigma_{\mathbf{u}}} + \overline{\Sigma_{\mathbf{u}'}} = r + kq$$

for some integer k and some $r \in [-4B, 4B]$. Therefore $\overline{\Sigma_{\mathbf{u}''}} = -r - kq$ and, as it is an element of $(-\frac{q}{2}, \frac{q}{2})$ (as any representative), this means that $\overline{\Sigma_{\mathbf{u}''}} = -r \in [-4B, 4B]$. Since the lemma assumes $\frac{q}{2} > 6B$, this means that $\overline{\Sigma_{\mathbf{u}''}} \notin (-\frac{q}{2}, -\frac{q}{2} + 2B] \cup [\frac{q}{2} - 2B, \frac{q}{2}]$. In other words, for all $\mathbf{w} \in \mathbb{Z}_q^{n-1}$, among the 3 representatives of any element of T, at least one is such that $\overline{\sum_j u_j w_j}$ is not in this set, which proves that $x \leq \frac{2}{3}$ and hence our lemma.

□

E Optimizing the Zero-Knowledge Framework

We detail here three independent optimizations of the framework from [YAZ+19]. We note that the first two optimizations apply as is to the original framework, while the third involves further changes.

Concrete Hardness Assumptions. The first one consists in changing the underlying hardness assumptions. Instead of using worst-case to average-case connections to standard lattice problems, we use slightly overstretched parameters for which the hardness of LWE is only based on concrete hardness arguments. The goal is to change the distribution of the randomness used to commit to the witness $\mathbf{x}$ in the original proof so that it leads to smaller elements. More precisely, we sample the randomness from a ternary distribution instead of a discrete Gaussian. Additionally, we add an extra verification step in order to rely on the HNF-SIS problem with two norm bounds β_∞, β_2 on the ℓ_∞ norm and ℓ_2 norm respectively (Definition 2.3). Again, now relying on concrete hardness arguments, we obtain an improved condition on q only depending on β_∞, which is usually much smaller than β_2. Moreover, as discussed after Definition 2.3, constraining the magnitude of the solution's coefficients seems to be beneficial for both theoretical and concrete hardness.

Rejection Sampling. The second modification we make is to better leverage the rejection sampling result from Corollary E.1 adapted from [Lyu12]. This step ensures that a discrete Gaussian sample shifted by a small enough vector is statistically close to the original discrete Gaussian distribution, thus masking the shift. However, this fact is not used to its full potential in the proof of [YAZ+19]. Doing so leads to smaller bounds in the verification equations and SIS norm bounds as a result. Additionally, we note when computing the size of the proof (in the non-interactive version), the authors treat the discrete Gaussian vectors as mere vectors over $\mathbb{Z}_q$. We can thus reduce the amount of storage needed as they have small coefficients with overwhelming probability, which results in smaller proofs by up to 20%.

Compacted Commitments. The final optimization regards the case when the proof system is run only once. It is sometimes better to increase the size p of the challenges rather than re-iterate the proof several times in order to achieve negligible soundness error. When running it once, we can compact the commitments thus limiting the number of elements to send and the size of the proof as a consequence. We note that compacting the commitments may not be desirable when the proof system is run multiple times as it would involve committing to the witness $\mathbf{x}$ multiple times.

E.1 Preliminaries

In what follows we sample the commitment randomness from a small distribution in order to optimize the parameters and the efficiency. For that, we employ the following ternary distribution which we denote by ψ_1, instead of Gaussian distributions. It outputs 0 with probability 6/16 and $-1, 1$ both with probability 5/16. This distribution has the advantage of being very efficiently sampleable as it only requires the sampling of 4 uniformly random bits to output a sample of ψ_1. For $\mathbf{x}$ sampled from ψ_1^n, it holds that $\|\mathbf{x}\|_2^2$ is distributed according to a binomial distribution with parameter $(n, 5/8)$. As such, Hoeffding's inequality gives the following.

Lemma E.1. *Let n be a positive integer. Then for all $\delta > 0$ it holds*

$$\mathbb{P}_{\mathbf{x} \hookleftarrow \psi_1^n}\left[\|\mathbf{x}\|_2 \geq \sqrt{(1+\delta)\frac{5}{8}n}\right] \leq \exp\left(-\frac{25}{32}\delta^2 n\right).$$

The above probability becomes 0 when $\delta > 3/5$.

Proof. Let $\mathbf{x}$ be a random vector whose coefficient are independent and identically distributed according to ψ_1. Therefore, for all $i \in [n]$, x_i^2 follows a Bernoulli distribution with parameter $5/8$. Then, define the random variable $X = \|\mathbf{x}\|_2^2 = \sum_{i\in[n]} x_i^2$. Hence, since X follows a binomial distribution $\mathcal{B}(n, 5/8)$. By Hoeffding's inequality, for all $t > 0$, we have

$$\mathbb{P}[X - \mathbb{E}[X] \geq t] \leq e^{-2t^2/n}.$$

Since $\mathbb{E}[X] = 5n/8$, then it holds that for all $\delta > 0$, setting $t = 5n\delta/8 > 0$ gives

$$\mathbb{P}[X \geq (1+\delta)5n/8] \leq e^{-25\delta^2 n/32}.$$

Therefore, it holds

$$\forall \delta > 0, \mathbb{P}_{\mathbf{x}\hookleftarrow\psi_1^n}\left[\|\mathbf{x}\|_2 \geq \sqrt{(1+\delta)\frac{5}{8}n}\right] \leq \exp(-\frac{25}{32}\delta^2 n).$$

□

We also recall the rejection sampling results from [Lyu12, Thm. 4.6, Lem. 4.7], which we adapt slightly to our case. In particular, we allow for a probabilistic bound on the shifts from V, resulting in very small changes in the acceptance probability and statistical distance.

Lemma E.2 (Adapted from [Lyu12, Thm. 4.6, Lem. 4.7]). *Let n be a positive integer, and V, Z two countable set of $\mathbb{R}^n$. Let T be a positive real, and we define $V_T = \{\mathbf{v} \in V : \|\mathbf{v}\|_2 \leq T\}$. Let h be a probability distributions on V such that $\mathbb{P}_{\mathbf{v}\sim h}[\mathbf{v} \notin V_T] \leq \varepsilon'$ for some $\varepsilon' \geq 0$. Let f be a probability distribution on Z, and $(g_{\mathbf{v}})_{\mathbf{v}\in V}$ a family of probability distributions on Z such that*

$$\exists M > 0, \forall \mathbf{v} \in V_T, \mathbb{P}_{\mathbf{z}\sim f}[M g_{\mathbf{v}}(\mathbf{z}) \geq f(\mathbf{z})] \geq 1 - \varepsilon,$$

for some $\varepsilon \geq 0$. We then define two distributions

$\mathcal{P}_1$**:** *Sample $\mathbf{v} \hookleftarrow h$, and $\mathbf{z} \hookleftarrow g_{\mathbf{v}}$. Output $(\mathbf{v}, \mathbf{z})$ with probability* $\min(1, \frac{f(\mathbf{z})}{Mg_{\mathbf{v}}(\mathbf{z})})$.
$\mathcal{P}_2$**:** *Sample $\mathbf{v} \hookleftarrow h$, and $\mathbf{z} \hookleftarrow f$. Output $(\mathbf{v}, \mathbf{z})$ with probability $1/M$.*

Then, it holds that $\mathcal{P}_1$ outputs something with probability at least $\frac{(1-\varepsilon)(1-\varepsilon')}{M}$, and that

$$\varDelta(\mathcal{P}_1, \mathcal{P}_2) \leq \max\left(\varepsilon' + \frac{\varepsilon+\varepsilon'}{2M}, \frac{\varepsilon}{M} + \frac{\varepsilon'}{2}\left(1 + \frac{1-\varepsilon}{M}\right)\right).$$

When $\varepsilon = \varepsilon'$ and $M \geq 1$, we simply have $\varDelta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon(1 + 1/M)$.

Proof. For each $\mathbf{v} \in V_T$, we define $S_{\mathbf{v}} = \{\mathbf{z} \in Z : Mg_{\mathbf{v}}(\mathbf{z}) \geq f(\mathbf{z})\}$. We first bound the probability that $\mathcal{P}_1$ outputs something, where the probability is over all the randomness. We denote this event by E_1. We have the following

$$\begin{aligned}
\mathbb{P}[E_1] &= \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}')\mathbb{P}[E_1 | \mathbf{v} = \mathbf{v}'] + \sum_{\mathbf{v}' \notin V_T} h(\mathbf{v}')\mathbb{P}[E_1 | \mathbf{v} = \mathbf{v}'] \\
&\geq \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \sum_{\mathbf{z} \in Z} g_{\mathbf{v}'}(\mathbf{z}) \cdot \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}'}(\mathbf{z})}, 1\right) \\
&= \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \left(\sum_{\mathbf{z} \in S_{\mathbf{v}'}} \frac{f(\mathbf{z})}{M} + \sum_{\mathbf{z} \notin S_{\mathbf{v}'}} g_{\mathbf{v}'}(\mathbf{z})\right) \\
&\geq \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \sum_{\mathbf{z} \in S_{\mathbf{v}'}} \frac{f(\mathbf{z})}{M} \\
&= \frac{1}{M} \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \mathbb{P}_{\mathbf{z} \sim f}[M g_{\mathbf{v}'}(\mathbf{z}) \geq f(\mathbf{z})] \\
&\geq \frac{1-\varepsilon}{M} \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \\
&= \frac{1-\varepsilon}{M} \mathbb{P}_{\mathbf{v} \sim f}[\mathbf{v} \in V_T] \\
&\geq \frac{(1-\varepsilon)(1-\varepsilon')}{M}.
\end{aligned}$$

Further, we also need to lower bound $\mathbb{P}[E_1]$. We rely on the fact that $\sum_{\mathbf{v}' \notin V_T} h(\mathbf{v}') = \mathbb{P}_{\mathbf{v} \sim h}[\mathbf{v} \notin V_T] \leq \varepsilon'$. We thus get

$$\begin{aligned}
\mathbb{P}[E_1] &= \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}')\mathbb{P}[E_1 | \mathbf{v} = \mathbf{v}'] + \sum_{\mathbf{v}' \notin V_T} h(\mathbf{v}')\mathbb{P}[E_1 | \mathbf{v} = \mathbf{v}'] \\
&\leq \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}')\mathbb{P}[E_1 | \mathbf{v} = \mathbf{v}'] + \sum_{\mathbf{v}' \notin V_T} h(\mathbf{v}') \\
&\leq \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \left(\sum_{\mathbf{z} \in S_{\mathbf{v}'}} \frac{f(\mathbf{z})}{M} + \sum_{\mathbf{z} \notin S_{\mathbf{v}'}} g_{\mathbf{v}'}(\mathbf{z})\right) + \varepsilon' \\
&\leq \sum_{\mathbf{v}' \in V_T} h(\mathbf{v}') \left(\sum_{\mathbf{z} \in S_{\mathbf{v}'}} \frac{f(\mathbf{z})}{M} + \sum_{\mathbf{z} \notin S_{\mathbf{v}'}} \frac{f(\mathbf{z})}{M}\right) + \varepsilon' \\
&= \frac{1}{M} \mathbb{P}_{\mathbf{v} \sim h}[\mathbf{v} \in V_T] + \varepsilon' \\
&\leq \frac{1}{M} + \varepsilon'.
\end{aligned}$$

Next, let p_i be the probability that $\mathcal{P}_i$ does not output anything. We have proven that $p_1 \in [1 - 1/M - \varepsilon', 1 - (1-\varepsilon)(1-\varepsilon')/M]$, and we trivially have $p_2 = 1 - 1/M$.

Meanwhile, we have

$$\begin{aligned}
\mathit{\Delta}(\mathcal{P}_1, \mathcal{P}_2) &= \frac{|p_1 - p_2|}{2} + \frac{1}{2}\sum_{\mathbf{v}\in V}\sum_{\mathbf{z}\in Z}|\mathcal{P}_1(\mathbf{v},\mathbf{z}) - \mathcal{P}_2(\mathbf{v},\mathbf{z})| \\
&\leq \frac{|p_1 - p_2|}{2} + \frac{1}{2}\sum_{\mathbf{v}\in V} h(\mathbf{v}) \sum_{\mathbf{z}\in Z}\left| g_{\mathbf{v}}(\mathbf{z}) \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right) - \frac{f(\mathbf{z})}{M}\right| \\
&= \frac{|p_1 - p_2|}{2} + \frac{1}{2}\sum_{\mathbf{v}\in V_T} h(\mathbf{v}) \sum_{\mathbf{z}\in Z}\left| g_{\mathbf{v}}(\mathbf{z}) \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right) - \frac{f(\mathbf{z})}{M}\right| \\
&\quad + \frac{1}{2}\sum_{\mathbf{v}\notin V_T} h(\mathbf{v}) \sum_{\mathbf{z}\in Z}\left| g_{\mathbf{v}}(\mathbf{z}) \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right) - \frac{f(\mathbf{z})}{M}\right|.
\end{aligned}$$

We then bound the two sums separately. First, for all $\mathbf{v}$ in V_T, it holds that

$$\begin{aligned}
\sum_{\mathbf{z}\in Z}\left| g_{\mathbf{v}}(\mathbf{z}) \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right) - \frac{f(\mathbf{z})}{M}\right| &= \sum_{\mathbf{z}\in S_{\mathbf{v}}}\left|\frac{f(\mathbf{z})}{M} - \frac{f(\mathbf{z})}{M}\right| + \sum_{\mathbf{z}\notin S_{\mathbf{v}}}\left| g_{\mathbf{v}}(\mathbf{z}) - \frac{f(\mathbf{z})}{M}\right| \\
&= \sum_{\mathbf{z}\notin S_{\mathbf{v}}} \frac{f(\mathbf{z})}{M} - g_{\mathbf{v}}(\mathbf{z}) \\
&\leq \sum_{\mathbf{z}\notin S_{\mathbf{v}}} \frac{f(\mathbf{z})}{M} \\
&= \frac{1}{M}\mathbb{P}_{\mathbf{z}\sim f}[\mathbf{z}\notin S_{\mathbf{v}}] \\
&\leq \frac{\varepsilon}{M}.
\end{aligned}$$

This means that the first sum can be bounded by $\frac{\varepsilon}{2M}\mathbb{P}_{\mathbf{v}\sim h}[\mathbf{v}\in V_T] \leq \frac{\varepsilon}{2M}$. To bound the second sum, we use the triangle inequality and for all $\mathbf{v}$ in $V \setminus V_T$ we get that

$$\sum_{\mathbf{z}\in Z}\left| g_{\mathbf{v}}(\mathbf{z}) \min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right) - \frac{f(\mathbf{z})}{M}\right| \leq g_{\mathbf{v}}(Z) + \frac{f(Z)}{M} = 1 + \frac{1}{M},$$

as the minimum in the sum can be bounded by 1. As a result, the second sum can be bounded above by $\frac{1}{2}(1 + \frac{1}{M})\mathbb{P}_{\mathbf{v}\sim h}[\mathbf{v}\notin V_T] \leq \frac{\varepsilon'(M+1)}{2M}$. Finally, from the bounds derived on p_1, we have $|p_1 - p_2| \leq \max(\varepsilon', \varepsilon(1-\varepsilon')/M)$. We thus get

$$\begin{aligned}
\mathit{\Delta}(\mathcal{P}_1, \mathcal{P}_2) &\leq \frac{\varepsilon + \varepsilon'(M+1)}{2M} + \max\left(\frac{\varepsilon'}{2}, \frac{\varepsilon(1-\varepsilon')}{M}\right) \\
&= \max\left(\varepsilon' + \frac{\varepsilon + \varepsilon'}{2M}, \frac{\varepsilon}{M} + \frac{\varepsilon'}{2}\left(1 + \frac{1-\varepsilon}{M}\right)\right).
\end{aligned}$$

It is easy to verify that when $M \geq 1$ and $\varepsilon' = \varepsilon$, the first term in the maximum above is the largest, and equals $\varepsilon(1 + 1/M)$. □

We then obtain the following corollary. We formulate it to give the freedom to choose the repetition rate M and the tail bound error. Note that in [Lyu12], M is determined by T, σ and the tail bound error. We instead choose M and the tail bound error, and then determine the minimal σ needed.

Corollary E.1. *Let n, p be positive integers. Let $\mathcal{L}$ be a lattice of rank n, and let $V = [-p, p]^n$. Let $T = p\sqrt{5n(1+\delta)/8}$, where $\delta = \sqrt{32(\lambda+1)/(25n \log_2 e)}$. Define h the distribution obtained by sampling α from $U([-p, p])$ and $\mathbf{s}$ from ψ_1^n and outputting $\mathbf{v} = \alpha\mathbf{s}$. Further, let $M > 1$, $t = \sqrt{(\lambda+2)/(\pi \log_2 e)}$ and define $\sigma_{\min} = (-t + \sqrt{t^2 + \ln(M)/\pi})^{-1} \cdot T$. Let $\sigma \geq \sigma_{\min}$. We now define two distributions*

- $\mathcal{P}_1$**:** *Sample $\mathbf{v} \hookleftarrow h$ and $\mathbf{y} \hookleftarrow \mathcal{D}_{\mathcal{L},\sigma}$. Define $\mathbf{z} = \mathbf{y} + \mathbf{v}$. Output $(\mathbf{v}, \mathbf{z})$ with probability $\min(1, \frac{\mathcal{D}_{\mathcal{L},\sigma}(\mathbf{z})}{M \cdot \mathcal{D}_{\mathcal{L},\sigma}(\mathbf{z}-\mathbf{v})})$.*
- $\mathcal{P}_2$**:** *Sample $\mathbf{v} \hookleftarrow h$ and $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L},\sigma}$. Output $(\mathbf{v}, \mathbf{z})$ with probability $1/M$.*

Then, it holds that $\mathcal{P}_1$ outputs something with probability at least $(1 - 2^{-\lambda})/M$, and that $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq 2^{-(\lambda+1)}(1 + 1/M) \leq 2^{-\lambda}$.

Proof. Due to the definition of h, T, δ and Lemma E.1, we have $\mathbb{P}_{\mathbf{v}\sim h}[\|\mathbf{v}\|_2 > T] \leq \exp(-\frac{25}{32}\delta^2 n) = 2^{-\lambda-1}$. Using the notations from Lemma E.2, we set $\varepsilon' = 2^{-\lambda-1}$. Now let $f = \mathcal{D}_{\mathcal{L},\sigma}$ and $(g_{\mathbf{v}})_{\mathbf{v}\in V} = (\mathbf{v} + \mathcal{D}_{\mathcal{L},\sigma})_{\mathbf{v}\in V}$. We simply have to verify that

$$\forall \mathbf{v} \in V_T, \mathbb{P}_{\mathbf{z}\sim f}[M g_{\mathbf{v}}(\mathbf{z}) \geq f(\mathbf{z})] \geq 1 - 2^{-\lambda-1}. \tag{13}$$

Let $\mathbf{v} \in V_T$, and $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L},\sigma}$. Then, we have

$$\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} = \frac{\mathcal{D}_{\mathcal{L},\sigma}(\mathbf{z})}{\mathcal{D}_{\mathcal{L},\sigma}(\mathbf{z}-\mathbf{v})} = \exp\left(\frac{\pi}{\sigma^2}(\|\mathbf{v}\|_2^2 - 2\langle \mathbf{v}, \mathbf{z}\rangle)\right).$$

Except with probability at most $2e^{-\pi t^2} = 2^{-\lambda-1}$, it holds that $|\langle \mathbf{v}, \mathbf{z}\rangle| \leq \sigma t \|\mathbf{v}\|_2$ by Lemma 2.3. We now condition on $|\langle \mathbf{v}, \mathbf{z}\rangle| \leq \sigma t \|\mathbf{v}\|_2$. It yields

$$\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} \leq \exp\left(\frac{\pi}{\sigma^2}(\|\mathbf{v}\|_2^2 + 2\sigma t\|\mathbf{v}\|_2)\right) \leq \exp\left(\pi((T/\sigma)^2 + 2t(T/\sigma))\right).$$

The way we defined $\sigma_{\min}$, we have that $T/\sigma_{\min}$ is the only positive solution to $x^2 + 2tx - \ln(M)/\pi = 0$. Since, we have $\sigma \geq \sigma_{\min}$, we have that T/σ is between the two solutions of the equation and as such we have that $(T/\sigma)^2 + 2t(T/\sigma) - \ln(M)/\pi \leq 0$. It can be re-written as

$$\exp\left(\pi((T/\sigma)^2 + 2t(T/\sigma))\right) \leq M,$$

Hence, conditioned on $|\langle \mathbf{v},\mathbf{z}\rangle| \leq \sigma t\|\mathbf{v}\|_2$, it holds that $f(\mathbf{z})/g_{\mathbf{v}}(\mathbf{z}) \leq M$. We obtain

$$\begin{aligned}\mathbb{P}_{\mathbf{z}\sim\mathcal{D}_{\mathcal{L},\sigma}}\left[\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} \leq M\right] &= \mathbb{P}_{\mathbf{z}}[|\langle \mathbf{v},\mathbf{z}\rangle| \leq \sigma t\|\mathbf{v}\|_2]\mathbb{P}_{\mathbf{z}}\left[\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} \leq M \middle| |\langle \mathbf{v},\mathbf{z}\rangle| \leq \sigma t\|\mathbf{v}\|_2\right] \\ &+ \mathbb{P}_{\mathbf{z}}[|\langle \mathbf{v},\mathbf{z}\rangle| > \sigma t\|\mathbf{v}\|_2]\mathbb{P}_{\mathbf{z}}\left[\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} \leq M \middle| |\langle \mathbf{v},\mathbf{z}\rangle| > \sigma t\|\mathbf{v}\|_2\right] \\ &\geq (1-2^{-\lambda-1})\mathbb{P}_{\mathbf{z}}\left[\frac{f(\mathbf{z})}{g_{\mathbf{v}}(\mathbf{z})} \leq M \middle| |\langle \mathbf{v},\mathbf{z}\rangle| \leq \sigma t\|\mathbf{v}\|_2\right] \\ &= 1-2^{-\lambda-1}.\end{aligned}$$

thus proving Equation (13) as required. Then, we can set $\varepsilon = 2^{-\lambda-1} = \varepsilon'$. Invoking Lemma E.2 yields the result. The probability of outputting something is at least $(1-\varepsilon)^2/M \geq (1-2\varepsilon)/M = (1-2^{-\lambda})/M$. □

The security properties of the zero-knowledge argument rely on the *Short Integer Solution* (SIS) problem [Ajt96] and the *Learning With Errors* (LWE) problem [Reg05] in Hermite Normal Form (HNF).

Definition E.1 ((HNF) Short Integer Solution). *Let n, m, q be positive integers, and $\beta_2 \geq \beta_\infty \geq 1$. The* Hermite Normal Form Short Integer Solution *problem, denoted by* $\text{HNF-SIS}^{\infty,2}_{n,m,q,\beta_\infty,\beta_2}$*, consists in finding $\mathbf{x} \in \mathcal{L}_q^\perp([\mathbf{I}_n|\mathbf{A}'])$ given $\mathbf{A}' \hookleftarrow U(\mathbb{Z}_q^{n\times(m-n)})$ such that $0 < \|\mathbf{x}\|_\infty \leq \beta_\infty$ and $0 < \|\mathbf{x}\|_2 \leq \beta_2$.*

We say that HNF-SIS is δ-hard if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the probability of $\mathcal{A}$ finding such a vector is at most δ over the randomness of $\mathbf{A}'$.

Definition E.2 ((HNF) Learning With Errors). *Let n, m, q be positive integers, and ψ a probability distribution over $\mathbb{Z}$. The Hermite Normal Form Learning With Errors problem, denoted by* $\text{HNF-LWE}_{n,m,q,\psi}$*, asks to distinguish between the following two distributions: (1) $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e} \bmod q)$ with $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\mathbf{s} \hookleftarrow \psi^n$ and $\mathbf{e} \hookleftarrow \psi^m$; (2) $(\mathbf{A}, \mathbf{b})$ with $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\mathbf{b} \hookleftarrow U(\mathbb{Z}_q^m)$.*

We say that HNF-LWE is δ-hard if for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in distinguishing both distributions is at most δ.

Finally, we briefly recall the security properties of a commitment scheme $\mathsf{aCommit}(m;\rho)$ which commits to a message m under randomness ρ. We say that $\mathsf{aCommit}$ is δ-hiding if a PPT adversary $\mathcal{A}$ has advantage at most δ in the following game: $\mathcal{A}$ chooses $m_0 \neq m_1$, receives $\mathsf{aCommit}(m_b;\rho)$ where b is a random bit, and outputs $b' \in \{0,1\}$. $\mathcal{A}$ wins if $b' = b$. We say that $\mathsf{aCommit}$ is δ-binding if a PPT adversary has advantage at most δ in outputting $(m_0,\rho_0),(m_1,\rho_1)$ such that $m_0 \neq m_1$ and $\mathsf{aCommit}(m_0;\rho_0) = \mathsf{aCommit}(m_1;\rho_1)$.

E.2 The Optimized Protocol.

We now present the main protocol with the optimizations we presented. Let ℓ_1, ℓ_2 be two positive integers. We denote by $L_{\mathbf{x}}$ the size of the witness vector, and $L_{\mathcal{M}}$

the size of the quadratic constraints set. We also define $L = \ell_1 + \ell_2 + L_{\mathbf{x}} + L_{\mathcal{M}}$. As is done in [YAZ$^+$19], we employ the homomorphic commitment scheme from [BDL$^+$18] over the integers. More precisely, we define

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{\ell_1} & & \mathbf{C}_1 \\ \mathbf{0}_{L_{\mathbf{x}}+L_{\mathcal{M}}\times\ell_1} & \mathbf{I}_{L_{\mathbf{x}}+L_{\mathcal{M}}} & \mathbf{C}_2 \end{bmatrix} \in \mathbb{Z}_q^{(\ell_1+L_{\mathbf{x}}+L_{\mathcal{M}})\times L},$$

with $\mathbf{C}_1 \hookleftarrow U(\mathbb{Z}_q^{\ell_1\times(L_{\mathbf{x}}+L_{\mathcal{M}}+\ell_2)})$, $\mathbf{C}_2 \hookleftarrow U(\mathbb{Z}_q^{(L_{\mathbf{x}}+L_{\mathcal{M}})\times\ell_2})$. Let $p = 2^\lambda$ be the maximal magnitude of the challenges, and $M > 1$ the repetition rate of the rejection sampling procedure. We then set δ, t, T as per Corollary E.1, namely $\delta = \sqrt{\frac{32}{25\log_2 e}\cdot\frac{\lambda+1}{L}}$, $t = \sqrt{(\lambda+2)/(\pi\log_2 e)}$ and $T = p\sqrt{5(1+\delta)/(8L)}$. Next define $s_2 = (-t + \sqrt{t^2 + \ln(M)/\pi})^{-1}\cdot T$. We define the rejection sampling function by $\mathfrak{p}(\mathbf{v},\mathbf{z}) = \min(1, \mathcal{D}_{\mathbb{Z}^L,s_2}(\mathbf{z})/(M\cdot\mathcal{D}_{\mathbb{Z}^L,s_2}(\mathbf{z}-\mathbf{v})))$ for all $\mathbf{v},\mathbf{z}\in\mathbb{Z}^L$. Finally, let $\mathsf{aCommit}$ be an auxiliary commitment scheme with randomness space $\{0,1\}^\kappa$ and message space $\mathbb{Z}_q^{k+2(\ell_1+L_{\mathbf{x}}+L_{\mathcal{M}})}$, and that is binding and hiding. The following interactive protocol involves a prover $\mathcal{P}$ with public input $\overline{\mathbf{A}}\in\mathbb{Z}_q^{k\times L_{\mathbf{x}}}$, $\mathbf{y}\in\mathbb{Z}_q^k$, and $\mathcal{M}\subseteq[L_{\mathbf{x}}]^3$ with $|\mathcal{M}| = L_{\mathcal{M}}$ and private input $\mathbf{x}\in\mathbb{Z}_q^{L_{\mathbf{x}}}$. The verifier $\mathcal{V}$ is only given the public input. In the protocol, $\mathcal{P}$ must convince $\mathcal{V}$ in zero-knowledge that they know $\mathbf{x}$ verifying

$$\begin{cases} \overline{\mathbf{A}}\mathbf{x} = \mathbf{y} \bmod q \\ \forall (h,i,j)\in\mathcal{M}, \mathbf{x}[h] = \mathbf{x}[i]\mathbf{x}[j] \bmod q \end{cases} \tag{14}$$

Theorem E.1. *The protocol described in Figure E.1 is complete with completeness error at most $\delta_c = 1 - 1/M + \mathsf{negl}(\lambda)$.*

We define $\beta_\infty = 8ps_2\log_2 L$ and $\beta_2 = 8ps_2\sqrt{L}$. Assume HNF-SIS$_{\ell_1,L,q,\beta_\infty,\beta_2}$ *is δ_{SIS}-hard and that* $\mathsf{aCommit}$ *is δ_b^a-binding. Then, there exists an extractor $\mathcal{E}$ that for any $\overline{\mathbf{A}},\mathbf{y},\mathcal{M}$ and any* PPT *cheating prover $\widehat{\mathcal{P}}$, if $\widehat{\mathcal{P}}$ can convince a verifier $\mathcal{V}$ without knowing a witness with probability at least $2/(2p+1)+\varepsilon$ for a non-negligible ε, then $\mathcal{E}$ can extract an $\mathbf{x}$ that verifies* (14) *in polynomial time, except with probability δ_{SIS}.*

Finally, assume that HNF-LWE$_{\ell_2,\ell_1+L_{\mathbf{x}}+L_{\mathcal{M}},q,\psi_1}$ *is δ_{LWE}-hard, and that the commitment* $\mathsf{aCommit}$ *is δ_h^a-hiding. Then, there exists a simulator $\mathcal{S}$ that with input $\overline{\mathbf{A}},\mathbf{y},\mathcal{M}$ outputs a transcript that is $(\delta_h^a + 2^{-\lambda-1}(1+1/M)+\delta_{\text{LWE}})$-indistinguishable from the transcript of an honest execution of the protocol with a prover knowing a witness $\mathbf{x}$ satisfying* (14).

Although the proof of Theorem E.1 follows naturally from that of [YAZ$^+$19], we give it in Section E.3. The above protocol can be turned into a non-interactive zero-knowledge argument of knowledge via the Fiat-Shamir heuristic in the random oracle model. In this case, the resulting proof does not contain the whole transcript as some elements are uniquely determined by the others for the proof to be correct. More precisely, the proof is $\pi = (\alpha, \rho, \mathbf{c}_1, \mathbf{z}_0, \mathbf{z}_1)$ where the challenge $\alpha = H(\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}, C_{aux}; AUX)$ with AUX an auxiliary input. The verification algorithm then re-computes $\mathbf{t}$ from the verification equation (4), $\mathbf{c}_2$ from

```
Prover $\mathcal{P}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}; \mathbf{x}]$                                        Verifier $\mathcal{V}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}]$
$\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^{L_\mathbf{x}})$
$\mathbf{t} \leftarrow \overline{\mathbf{A}}\mathbf{r} \bmod q$
$\mathbf{s}_1 \hookleftarrow \psi_1^L$
$\mathbf{s}_2 \hookleftarrow \mathcal{D}_{\mathbb{Z}^L, s_2}$
Let $\mathbf{a}, \mathbf{b}$ be $L_\mathcal{M}$-dimensional vectors
For $e \in [L_\mathcal{M}]$, let $(h, i, j)$ be the $e$-th element of $\mathcal{M}$
  $\mathbf{a}[e] \leftarrow \mathbf{r}[h] - \mathbf{r}[i]\mathbf{x}[j] - \mathbf{r}[j]\mathbf{x}[i]$
  $\mathbf{b}[e] \leftarrow -\mathbf{r}[i]\mathbf{r}[j]$
$\mathbf{c}_1 \leftarrow \mathbf{C}\mathbf{s}_1 + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ \mathbf{x} \\ \mathbf{a} \end{bmatrix} \bmod q$
$\mathbf{c}_2 \leftarrow \mathbf{C}\mathbf{s}_2 + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ \mathbf{r} \\ \mathbf{b} \end{bmatrix} \bmod q$
$\rho \hookleftarrow U(\{0,1\}^\kappa)$
$C_{aux} \leftarrow \mathsf{aCommit}(\mathbf{t} \| \mathbf{c}_1 \| \mathbf{c}_2; \rho)$
                              $\xrightarrow{C_{aux}}$
                                                  $\alpha \hookleftarrow U([-p, p])$
                              $\xleftarrow{\alpha}$
$\mathbf{z}_0 \leftarrow \alpha\mathbf{x} + \mathbf{r}$
$\mathbf{z}_1 \leftarrow \alpha\mathbf{s}_1 + \mathbf{s}_2$
Abort with probability $1 - \mathfrak{p}(\alpha\mathbf{s}_1, \mathbf{z}_1)$
                              $\xrightarrow{\mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \rho, \mathbf{z}_0, \mathbf{z}_1}$
                                            Let $\mathbf{d}$ be an $L_\mathcal{M}$-dimensional vector
                                            For $e \in [L_\mathcal{M}]$, let $(h, i, j)$ be the $e$-th element of $\mathcal{M}$
                                              $\mathbf{d}[e] \leftarrow \alpha\mathbf{z}_0[h] - \mathbf{z}_0[i]\mathbf{z}_0[j]$
                                            Accept if:
                                              (1) $C_{aux} = \mathsf{aCommit}(\mathbf{t} \| \mathbf{c}_1 \| \mathbf{c}_2; \rho)$
                                              (2) $\|\mathbf{z}_1\|_\infty \leq s_2 \log_2 L$
                                              (3) $\|\mathbf{z}_1\|_2 \leq s_2\sqrt{L}$
                                              (4) $\overline{\mathbf{A}}\mathbf{z}_0 = \alpha\mathbf{y} + \mathbf{t} \bmod q$
                                              (5) $\mathbf{C}\mathbf{z}_1 + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ \mathbf{z}_0 \\ \mathbf{d} \end{bmatrix} = \alpha\mathbf{c}_1 + \mathbf{c}_2 \bmod q$
```

Fig. E.1. Zero-knowledge Argument of Knowledge for Equation (14) with compacted commitments.

equation (5) and C_{aux} from equation (1). We end up with a proof of size

$$
\begin{aligned}
|\pi| = &\lceil \log_2(2p+1) \rceil + \kappa + (\ell_1 + L_\mathbf{x} + L_\mathcal{M})\lceil \log_2 q \rceil + L_\mathbf{x}\lceil \log_2 q \rceil \\
&+ L\lceil \log_2(s_2 \log_2 L) \rceil \qquad (15) \\
= &\lceil \log_2(2p+1) \rceil + \kappa + (\ell_1 + 2L_\mathbf{x} + L_\mathcal{M})\lceil \log_2 q \rceil + L\lceil \log_2(s_2 \log_2 L) \rceil \qquad (16)
\end{aligned}
$$

The last term in (15) does not appear in the proof size of [YAZ+19] as they treat $\mathbf{z}_1$ (and $\mathbf{z}_2$ in their case) as vectors in $\mathbb{Z}_q$. However, due to the rejection sampling, one has that they are Gaussian vectors and we can therefore reduce the amount of storage needed. Depending on the chosen parameters, this simple observation reduces the proof size by up to 20%.

E.3 Proof of Theorem E.1

Proof. Completeness: Consider an honest execution of the protocol, i.e., between a prover $\mathcal{P}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}; \mathbf{x}]$ with $\mathbf{x}$ satisfying (14), and a verifier $\mathcal{V}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}]$. Since the execution is honest and since $\mathsf{aCommit}$ does not use any internal randomness other than ρ, (1) is trivially verified. Next, due to the rejection sampling, $\mathcal{P}$ respond in the third move only with probability $\mathfrak{p}(\alpha\mathbf{s}_1, \mathbf{z}_1)$. By Corollary E.1, it holds that the prover responds with probability at least $(1-2^{-\lambda})/M$, and that $\mathbf{z}_1$ is within statistical distance $2^{-\lambda-1}(1+1/M)$ of $\mathcal{D}_{\mathbb{Z}^L, s_2}$. We further condition on a non-aborting transcript. Lemma 2.3 combined with the union bound gives

$$\mathbb{P}[\|\mathbf{z}_1\|_\infty > s_2 \log_2 L \vee \|\mathbf{z}_1\|_2 > s_2\sqrt{L}] \leq 2^{-\lambda-1}\left(1 + \frac{1}{M}\right) + 2^{-2L} + 2Le^{-\pi \log_2^2 L}.$$

Equation (4) is easily verified as $\overline{\mathbf{A}}\mathbf{z}_0 = \overline{\mathbf{A}}(\alpha\mathbf{x}+\mathbf{r}) = \alpha(\overline{\mathbf{A}}\mathbf{x}) + \overline{\mathbf{A}}\mathbf{r} = \alpha\mathbf{y}+\mathbf{t} \bmod q$. Now, let $e \in [L_\mathcal{M}]$ and let (h, i, j) be the e-th element of $\mathcal{M}$. We have

$$\begin{aligned}
\mathbf{d}[e] &= \alpha\mathbf{z}_0[h] - \mathbf{z}_0[i]\mathbf{z}_0[j] \\
&= \alpha(\alpha\mathbf{x}[h] + \mathbf{r}[h]) - (\alpha\mathbf{x}[i] + \mathbf{r}[i])(\alpha\mathbf{x}[j] + \mathbf{r}[j]) \\
&= \alpha^2(\mathbf{x}[h] - \mathbf{x}[i]\mathbf{x}[j]) + \alpha(\mathbf{r}[h] - \mathbf{r}[i]\mathbf{x}[j] - \mathbf{r}[j]\mathbf{x}[i]) + (-\mathbf{r}[i]\mathbf{r}[j]) \\
&= \alpha\mathbf{a}[e] + \mathbf{b}[e] \bmod q.
\end{aligned}$$

As a result, it holds that $\mathbf{d} = \alpha\mathbf{a} + \mathbf{b} \bmod q$. It thus yields

$$\begin{aligned}
\mathbf{C}\mathbf{z}_1 + \begin{bmatrix}\mathbf{0}_{\ell_1} \\ \mathbf{z}_0 \\ \mathbf{d}\end{bmatrix} &= \mathbf{C}(\alpha\mathbf{s}_1 + \mathbf{s}_2) + \begin{bmatrix}\mathbf{0}_{\ell_1} \\ \alpha\mathbf{x}+\mathbf{r} \\ \alpha\mathbf{a}+\mathbf{b}\end{bmatrix} \bmod q \\
&= \alpha\left(\mathbf{C}\mathbf{s}_1 + \begin{bmatrix}\mathbf{0}_{\ell_1} \\ \mathbf{x} \\ \mathbf{a}\end{bmatrix}\right) + \left(\mathbf{C}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0}_{\ell_1} \\ \mathbf{r} \\ \mathbf{b}\end{bmatrix}\right) \bmod q \\
&= \alpha\mathbf{c}_1 + \mathbf{c}_2 \bmod q,
\end{aligned}$$

proving (5). Combining it all yields

$$\mathbb{P}[\langle \mathcal{P}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}; \mathbf{x}], \mathcal{V}[\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}]\rangle \neq 1] \leq 1 - 1/M + \mathsf{negl}(\lambda).$$

Extractor: Now, assume that a cheating prover $\widehat{\mathcal{P}}$ can convince the verifier that they possess a witness for $(\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M})$ with probability $2/(2p+1) + \varepsilon$ for some non-negligible ε. We construct the extractor $\mathcal{E}$ that uses $\widehat{\mathcal{P}}$ via black-box access. First, $\mathcal{E}$ runs $\widehat{\mathcal{P}}$ until it obtains an accepting transcript $(\mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \alpha, \mathbf{z}_0, \mathbf{z}_1)$. Between each run, $\mathcal{E}$ rewinds the inner randomness of $\widehat{\mathcal{P}}$ to have the same first move response. This first transcript is obtained in expected time $T_1 = (2/(2p+1) + \varepsilon)^{-1}$. Then, $\mathcal{E}$ re-iterates the same process but sends challenges $\alpha' \neq \alpha$ to $\widehat{\mathcal{P}}$. Assuming $\mathsf{aCommit}$ is δ_b^a-binding and since the first move always uses the same randomness, $\mathcal{E}$ can therefore obtain another accepting transcript $(\mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \alpha', \mathbf{z}_0', \mathbf{z}_1')$

in expected time $T_2 = (1/(2p+1) + \varepsilon - \delta_b^a)^{-1}$. It then continues running $\widehat{\mathcal{P}}$ with challenges $\alpha'' \notin \{\alpha, \alpha'\}$ to get a third accepting transcript $(\mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \alpha'', \mathbf{z}_0'', \mathbf{z}_1'')$ in expected time $T_3 = (\varepsilon - \delta_b^a)^{-1}$. The total expected time is therefore $T = T_1 + T_2 + T_3 \leq \mathsf{poly}(\lambda)$. Finally, the extractor $\mathcal{E}$ outputs the witness $\overline{\mathbf{x}} = (\alpha' - \alpha)^{-1}(\mathbf{z}_0' - \mathbf{z}_0) \bmod q$. We now analyze the correctness of $\mathcal{E}$. We further define $\varDelta_1 = \alpha' - \alpha$ and $\varDelta_2 = \alpha'' - \alpha$. First, we have

$$\begin{aligned}
\overline{\mathbf{A}}\overline{\mathbf{x}} &= \varDelta_1^{-1}(\overline{\mathbf{A}}\mathbf{z}_0' - \overline{\mathbf{A}}\mathbf{z}_0) \bmod q \\
&= \varDelta_1^{-1}(\alpha'\mathbf{y} + \mathbf{t} - (\alpha\mathbf{y} + \mathbf{t})) \bmod q \text{ (by (4))} \\
&= \varDelta_1^{-1}(\alpha' - \alpha)\mathbf{y} \bmod q \\
&= \mathbf{y} \bmod q.
\end{aligned}$$

We now prove that $\overline{\mathbf{x}}$ verifies the quadratic constraints except with probability δ_{SIS}. For that, we define $\mathbf{e}' = \mathbf{z}_1' - \mathbf{z}_1$, $\mathbf{e}'' = \mathbf{z}_1'' - \mathbf{z}_1$, $\mathbf{f}' = \mathbf{z}_0' - \mathbf{z}_0$, $\mathbf{f}'' = \mathbf{z}_0'' - \mathbf{z}_0$, and $\mathbf{g}' = \mathbf{d}' - \mathbf{d}$, $\mathbf{g}'' = \mathbf{d}'' - \mathbf{d}$. The verification equation (5) gives

$$\begin{cases}
\mathbf{C}\mathbf{e}' + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ \mathbf{f}' \\ \mathbf{g}' \end{bmatrix} = \varDelta_1 \mathbf{c}_1 \bmod q \\
\mathbf{C}\mathbf{e}'' + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ \mathbf{f}'' \\ \mathbf{g}'' \end{bmatrix} = \varDelta_2 \mathbf{c}_1 \bmod q.
\end{cases}$$

Cancelling the right-hand side provides us with

$$\mathbf{C}(\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}'') + \begin{bmatrix} \mathbf{0}_{\ell_1} \\ (\varDelta_2\mathbf{f}' - \varDelta_1\mathbf{f}'') \\ (\varDelta_2\mathbf{g}' - \varDelta_1\mathbf{g}'') \end{bmatrix} = \mathbf{0} \bmod q.$$

The first block then gives $[\mathbf{I}_{\ell_1} | \mathbf{C}_1](\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}'') = \mathbf{0} \bmod q$. Yet, we can bound the norms of $\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}''$ using the verification equations (2) and (3) and get $\|\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}''\|_\infty \leq 8ps_2 \log_2 L = \beta_\infty$ and $\|\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}''\|_2 \leq 8ps_2\sqrt{L} = \beta_2$. Since we assume that $\text{HNF-SIS}_{\ell_1, L, q, \beta_\infty, \beta_2}$ is δ_{SIS}-hard, then no PPT adversary can solve it with advantage more than δ_{SIS}. Hence, we get that $\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}'' = \mathbf{0}$ except with probability at most δ_{SIS}. We now condition on $\varDelta_2\mathbf{e}' - \varDelta_1\mathbf{e}'' = \mathbf{0}$. The second and third blocks in the above yields $\varDelta_2\mathbf{f}' = \varDelta_1\mathbf{f}'' \bmod q$ and $\varDelta_2\mathbf{g}' = \varDelta_1\mathbf{g}'' \bmod q$. We now define $\overline{\mathbf{r}} = \mathbf{z}_0 - \alpha\overline{\mathbf{x}} \bmod q$. Then

$$\mathbf{z}_0' - \alpha'\overline{\mathbf{x}} = \mathbf{z}_0' - \varDelta_1\overline{\mathbf{x}} - \alpha\overline{\mathbf{x}} = \mathbf{z}_0' - \mathbf{f}' - \alpha\overline{\mathbf{x}} \bmod q = \overline{\mathbf{r}} \bmod q \tag{17}$$

$$\begin{aligned}
\mathbf{z}_0'' - \alpha''\overline{\mathbf{x}} = \mathbf{z}_0'' - \varDelta_2\overline{\mathbf{x}} - \alpha\overline{\mathbf{x}} &= \mathbf{z}_0'' - \varDelta_2\varDelta_1^{-1}\mathbf{f}' - \alpha\overline{\mathbf{x}} \\
&= \mathbf{z}_0'' - \varDelta_2\varDelta_2^{-1}\mathbf{f}'' - \alpha\overline{\mathbf{x}} \bmod q \\
&= \overline{\mathbf{r}} \bmod q.
\end{aligned} \tag{18}$$

Now let $e \in [L_{\mathcal{M}}]$ and (h, i, j) be the e-th element of $\mathcal{M}$. We have

$$\begin{aligned}\mathbf{d}[e] &= \alpha \mathbf{z}_0[h] - \mathbf{z}_0[i]\mathbf{z}_0[j] \\ &= \alpha(\alpha\overline{\mathbf{x}}[h] + \overline{\mathbf{r}}[h]) - (\alpha\overline{\mathbf{x}}[i] + \overline{\mathbf{r}}[i])(\alpha\overline{\mathbf{x}}[j] + \overline{\mathbf{r}}[j]) \\ &= \alpha^2(\overline{\mathbf{x}}[h] - \overline{\mathbf{x}}[i]\overline{\mathbf{x}}[j]) + \alpha(\overline{\mathbf{r}}[h] - \overline{\mathbf{r}}[i]\overline{\mathbf{x}}[j] - \overline{\mathbf{r}}[j]\overline{\mathbf{x}}[i]) + (-\overline{\mathbf{r}}[i]\overline{\mathbf{r}}[j]) \\ &= \mathbf{c}[e]\alpha^2 + \mathbf{a}[e]\alpha + \mathbf{b}[e].\end{aligned}$$

Due to Equations (17) and (18), we also have $\mathbf{d}' = {\alpha'}^2\mathbf{c} + \alpha'\mathbf{a} + \mathbf{b}$ and $\mathbf{d}'' = {\alpha''}^2\mathbf{c} + \alpha''\mathbf{a} + \mathbf{b}$. Hence, since $\Delta_1^{-1}\mathbf{g}' = \Delta_2^{-1}\mathbf{g}'' \bmod q$, we obtain

$$(\alpha' + \alpha)\mathbf{c} + \mathbf{a} = (\alpha'' + \alpha)\mathbf{c} + \mathbf{a} \bmod q$$

which leads to $(\alpha'' - \alpha)\mathbf{c} = \mathbf{0} \bmod q$. Since $\alpha'' \neq \alpha'$ and q is prime, then $\alpha'' - \alpha' \in \mathbb{Z}_q^\times$ and therefore $\mathbf{c} = \mathbf{0} \bmod q$. This proves that for all $(h, i, j) \in \mathcal{M}$, $\overline{\mathbf{x}}[h] = \overline{\mathbf{x}}[i]\overline{\mathbf{x}}[j] \bmod q$. As a result, the output of $\mathcal{E}$ is correct except with probability at most δ_{SIS}.

Simulator: We construct the following simulator $\mathcal{S}$ that simulates the distribution of an honest transcript but only using the public inputs. It proceeds as follows

1. $\alpha \hookleftarrow U([-p, p])$
2. $\mathbf{z}_0 \hookleftarrow U(\mathbb{Z}_q^{L_{\mathbf{x}}})$
3. $\mathbf{t} \leftarrow \overline{\mathbf{A}}\mathbf{z}_0 - \alpha\mathbf{y} \bmod q$
4. For $e \in [L_{\mathcal{M}}]$, let (h, i, j) be the e-the element of $\mathcal{M}$. Then, $\mathbf{d}[e] \leftarrow \alpha\mathbf{z}_0[h] - \mathbf{z}_0[i]\mathbf{z}_0[j]$
5. $\mathbf{z}_1 \hookleftarrow \mathcal{D}_{\mathbb{Z}^L, s_2}$
6. $\mathbf{c}_1 \hookleftarrow U(\mathbb{Z}_q^{\ell_1 + L_{\mathbf{x}} + L_{\mathcal{M}}})$
7. $\mathbf{c}_2 \leftarrow \mathbf{C}\mathbf{z}_1 + \begin{bmatrix}\mathbf{0}_{\ell_1} \\ \mathbf{z}_0 \\ \mathbf{d}\end{bmatrix} - \alpha\mathbf{c}_1 \bmod q$
8. $\rho \hookleftarrow U(\{0,1\}^\kappa)$
9. $C_{\mathsf{aux}} \leftarrow \mathsf{aCommit}(\mathbf{t}\|\mathbf{c}_1\|\mathbf{c}_2; \rho)$
10. $C'_{\mathsf{aux}} \leftarrow \mathsf{aCommit}(\mathbf{0}; \rho)$
11. Output $(C_{aux}, \alpha, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \rho, \mathbf{z}_0, \mathbf{z}_1)$ with probability $1/M$ and $(C'_{aux}, \alpha, \perp)$ otherwise.

We now prove that the output of $\mathcal{S}$ is computationally indistinguishable from the transcript of an honest execution of the protocol. We proceed by game hopping.
Game G_0: This corresponds to an honest execution.
Game G_1: Here, the prover $\mathcal{P}$ retrieves the challenge α from the honest verifier by sending $\mathsf{aCommit}(\mathbf{0}; \rho)$ for some $\rho \hookleftarrow U(\{0,1\}^\kappa)$. It then rewinds the verifier to its initial state including its inner randomness. It then proceeds as follows:

1. $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^{L_{\mathbf{x}}})$
2. $\mathbf{t} \leftarrow \overline{\mathbf{A}}\mathbf{r} \bmod q$
3. $\mathbf{s}_1 \hookleftarrow \psi_1^L$
4. $\mathbf{s}_2 \hookleftarrow \mathcal{D}_{\mathbb{Z}^L, s_2}$

5. For $e \in [L_{\mathcal{M}}]$, let (h, i, j) be the e-the element of $\mathcal{M}$. Then, $\mathbf{a}[e] \leftarrow \mathbf{r}[h] - \mathbf{r}[i]\mathbf{x}[j] - \mathbf{r}[j]\mathbf{x}[i]$ and $\mathbf{b}[e] \leftarrow -\mathbf{r}[i]\mathbf{r}[j]$
6. $\mathbf{c}_1 \leftarrow \mathbf{C}\mathbf{s}_1 + \begin{bmatrix}\mathbf{0}_{\ell_1}\\ \mathbf{x}\\ \mathbf{a}\end{bmatrix} \bmod q$
7. $\mathbf{c}_2 \leftarrow \mathbf{C}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0}_{\ell_1}\\ \mathbf{r}\\ \mathbf{b}\end{bmatrix} \bmod q$
8. $\mathbf{z}_0 \leftarrow \alpha\mathbf{x} + \mathbf{r}$
9. $\mathbf{z}_1 \leftarrow \alpha\mathbf{s}_1 + \mathbf{s}_2$
10. Set the binary variable abort to 1 with probability $1 - \mathfrak{p}(\alpha\mathbf{s}_1, \mathbf{z}_1)$
11. $\rho \hookleftarrow U(\{0,1\}^{\kappa})$
12. $C_{aux} \leftarrow \mathsf{aCommit}(\mathbf{t}\|\mathbf{c}_1\|\mathbf{c}_2; \rho)$ and it sends C_{aux} to the verifier
13. When receiving α' from the verifier, the prover aborts if $\mathsf{abort} = 1$ and otherwise sends $(\mathbf{t}, \mathbf{c}_1, \mathbf{c}_2, \rho, \mathbf{z}_0, \mathbf{z}_1)$.

Game G_2: It is identical to G_1 except in the computation of $\mathbf{t}$ and $\mathbf{c}_2$. They are instead computed to verify equations (4) and (5) in the verification:

1. $\mathbf{t} \leftarrow \overline{\mathbf{A}}\mathbf{z}_0 - \alpha\mathbf{y} \bmod q$
2. For $e \in [L_{\mathcal{M}}]$, let (h, i, j) be the e-the element of $\mathcal{M}$. Then, $\mathbf{d}[e] \leftarrow \alpha\mathbf{z}_0[h] - \mathbf{z}_0[i]\mathbf{z}_0[j]$
3. $\mathbf{c}_2 \leftarrow \mathbf{C}\mathbf{z}_1 + \begin{bmatrix}\mathbf{0}_{\ell_1}\\ \mathbf{z}_0\\ \mathbf{d}\end{bmatrix} - \alpha\mathbf{c}_1 \bmod q$

Game G_3: It is identical to G_2 except for the computation of C_{aux}.

1. $C_{aux} \leftarrow \mathsf{aCommit}(\mathbf{0}; \rho)$ if $\mathsf{abort} = 1$ and $C_{aux} \leftarrow \mathsf{aCommit}(\mathbf{t}\|\mathbf{c}_1\|\mathbf{c}_2; \rho)$ otherwise.

Game G_4: It is identical to G_3 except in the computation of $\mathbf{z}_1$ and abort.

1. $\mathbf{z}_1 \hookleftarrow \mathcal{D}_{\mathbb{Z}^L, s_2}$
2. Set $\mathsf{abort} = 1$ with probability $1 - 1/M$ and 0 otherwise

Game G_5: It is identical to G_4 except in the computation of $\mathbf{c}_1$.

1. $\mathbf{c}_1 \hookleftarrow U(\mathbb{Z}_q^{\ell_1 + L_{\mathbf{x}} + L_{\mathcal{M}}})$

Game G_6: It is identical to G_5 except in the computation of $\mathbf{z}_0$

1. $\mathbf{z}_0 \hookleftarrow U(\mathbb{Z}_q^{L_{\mathbf{x}}})$

We now prove that each game is indistinguishable from the next. First, since the verifier $\mathcal{V}$ is honest, the challenge α' is fully determined by its inner randomness. As it is rewinded, we always have $\alpha' = \alpha$. All other variables are identically distributed, which gives

$$\varDelta(\mathsf{View}_{G_0}(\mathcal{V}), \mathsf{View}_{G_1}(\mathcal{V})) = 0. \tag{19}$$

By the completeness of the protocol, $\mathbf{t}$ and $\mathbf{c}_2$ are uniquely determined by the other variables and the verification equations (4) and (5). Thus

$$\varDelta(\mathsf{View}_{G_1}(\mathcal{V}), \mathsf{View}_{G_2}(\mathcal{V})) = 0. \tag{20}$$

Since $\mathsf{aCommit}$ is δ_h^a-hiding, it holds that a PPT adversary $\mathcal{A}$ can distinguish between games G_2 and G_3 with advantage at most δ_h^a.

$$|\mathbb{P}[\mathcal{A}(\mathsf{View}_{G_2}(\mathcal{V})) = 1] - \mathbb{P}[\mathcal{A}(\mathsf{View}_{G_3}(\mathcal{V})) = 1]| \leq \delta_h^a. \tag{21}$$

Then, by Corollary E.1, it directly holds that the computation of $\mathbf{z}_1$ and abort in G_4 is within statistical distance $2^{-\lambda-1}(1 + 1/M)$ of that of game G_3. Hence

$$\varDelta(\mathsf{View}_{G_3}(\mathcal{V}), \mathsf{View}_{G_4}(\mathcal{V})) \leq 2^{-\lambda-1}(1 + 1/M). \tag{22}$$

We then use the hiding property of the commitment scheme from [BDL+18] to argue that G_4 and G_5 are indistinguishable under the LWE assumption. The details are already provided in [YAZ+19]. More precisely, since we assume that $\text{HNF-LWE}_{\ell_2,\ell_1+L_{\mathbf{x}}+L_{\mathcal{M}},q,\psi_1}$ is δ_{LWE}-hard, then for any PPT adversary $\mathcal{A}$ we get

$$|\mathbb{P}[\mathcal{A}(\mathsf{View}_{G_4}(\mathcal{V})) = 1] - \mathbb{P}[\mathcal{A}(\mathsf{View}_{G_5}(\mathcal{V})) = 1]| \leq \delta_{\text{LWE}}. \tag{23}$$

In G_5, $\mathbf{z}_0 = \alpha\mathbf{x} + \mathbf{r}$ where $\mathbf{r}$ is uniform in $\mathbb{Z}_q^{L_{\mathbf{x}}}$ and independent of $\alpha\mathbf{x}$. Hence, $\mathbf{z}_0$ is also uniform in $\mathbb{Z}_q^{L_{\mathbf{x}}}$. Thus:

$$\varDelta(\mathsf{View}_{G_5}(\mathcal{V}), \mathsf{View}_{G_6}(\mathcal{V})) = 0. \tag{24}$$

Then, the distribution of the transcript in G_6 no longer depends on the witness $\mathbf{x}$ and is exactly the same as the output of $\mathcal{S}$. Combining Equations (19), (20), (21), (22), (23) and (24) yields

$$\left|\mathbb{P}[\mathcal{A}(\mathsf{View}_{G_0}(\mathcal{V})) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{S}(\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M})) = 1]\right| \leq \delta_h^a + 2^{-\lambda-1}(1+1/M) + \delta_{\text{LWE}},$$

as desired. □

F Instantiating the Protocols

In this section we show how to instantiate the different relations to be proven in zero-knowledge with the proof system of [YAZ+19]. More precisely, we need to have zero-knowledge arguments for the opening of an Ajtai commitment, and for the verification equation of our signature with efficient protocols. We detail both in Sections F.1 and F.2. For completeness and for performance comparison purposes, we also instantiate the construction Libert et al. [LLM+16] with the framework of [YAZ+19].

When concretely evaluating these proofs, we consider the optimizations proposed in Appendix E, consisting in compacting the commitments of the original framework and in better parameter selections, which leads to substantial efficiency improvements.

F.1 Proof of Commitment Opening.

Consider a prover with private input $\mathbf{m} \in \{0,1\}^{m_3}$ and $\mathbf{r}' \sim \mathcal{D}_{\mathbb{Z}^{m_1},\sigma_3}$, and public input pp, pk. Recall that by Lemma 2.3, we have $\|\mathbf{r}'\|_\infty \leq \sigma_3 \log_2 m_1$ with overwhelming probability. We can thus define $\alpha_3 = \lceil \sigma_3 \log_2 m_1 \rceil$ and assume that $\mathbf{r}' \in [-\alpha_3, \alpha_3]^{m_1}$. The prover wishes to prove that

$$\mathbf{A}\mathbf{r}' + \mathbf{D}\mathbf{m} = \mathbf{c} \bmod q \ \wedge \ \|\mathbf{r}'\|_\infty \leq \alpha_3 \ \wedge \ \mathbf{m} \in \{0,1\}^{m_1}.$$

We thus transform this relation into one that fits the Yang et al. framework. For that, we first define $\mathbf{a}_3 = \alpha_3 \mathbf{1}_{m_1}$. Next, we define $\mathbf{r}'' = \mathbf{r}' + \mathbf{a}_3 \in [0, 2\alpha_3]^{m_1}$. Let $k_{\alpha_3} = \lfloor \log_2 2\alpha_3 \rfloor + 1$ and define[11] $\mathbf{g}_{\alpha_3} = [\lfloor (2\alpha_3 + 2^{i-1})/2^i \rfloor]_{i \in [k_{\alpha_3}]} \in \mathbb{Z}^{1 \times k_{\alpha_3}}$, and $\mathbf{G}_{\alpha_3} = \mathbf{I}_{m_1} \otimes \mathbf{g}_{\alpha_3}$. We then denote by $\overline{\mathbf{r}'} \in \{0,1\}^{m_1 k_{\alpha_3}}$ a binary decomposition of $\mathbf{r}''$ along $\mathbf{g}_{\alpha_3}$, i.e., that verifies $\mathbf{r}'' = \mathbf{G}_{\alpha_3}\overline{\mathbf{r}'}$. Such a decomposition can be efficiently computed. It now suffices to prove the following

$$\mathbf{A}\mathbf{G}_{\alpha_3}\overline{\mathbf{r}'} + \mathbf{D}\mathbf{m} = \mathbf{c} + \mathbf{A}\mathbf{a}_3 \bmod q \ \wedge \ \overline{\mathbf{r}'} \in \{0,1\}^{m_1 k_{\alpha_3}} \ \wedge \ \mathbf{m} \in \{0,1\}^{m_3}.$$

By defining $\overline{\mathbf{A}} = [\mathbf{A}\mathbf{G}_{\alpha_3} | \mathbf{D}]$, $\mathbf{x} = [\overline{\mathbf{r}'}^T | \mathbf{m}^T]^T$, $\mathbf{y} = \mathbf{c} + \mathbf{A}\mathbf{a}_3$ and $\mathcal{M} = \{(i,i,i); i \in [m_1 k_{\alpha_3} + m_3]\}$, we have $((\overline{\mathbf{A}}, \mathbf{y}, \mathcal{M}); \mathbf{x}) \in \mathcal{R}^*$. The length of the witness is $L_\mathbf{x} = m_1 k_{\alpha_3} + m_3$, and the size of $\mathcal{M}$ is $L_\mathcal{M} = L_\mathbf{x}$. Note that since q is prime, the constraint $\mathbf{x}[i] = \mathbf{x}[i]^2 \bmod q$ indeed implies that $\mathbf{x}[i] \in \{0,1\}$.

When $m_1 \gg \lambda$, the *fast mode* of Section 4.2 compresses the size of the witness and the constraint set as we now prove that $\mathbf{H}\mathbf{r}'$ has coefficients bounded by $\sigma_3\sqrt{m_1}\log_2 \lambda$. It yields a witness of size $L_\mathbf{x} = m_1 + k(\lfloor \log_2(2\sigma_3\sqrt{m_1}\log_2\lambda)\rfloor + 1) + m_3$, with $k = \lambda/\log_2(9/5)$, and $L_\mathcal{M} = L_\mathbf{x} - m_1$.

F.2 Proof of Message-Signature Pair Possession.

Here, the prover has a private input $\mathbf{m} \in \{0,1\}^{m_3}$ and $(\tau, \mathbf{v}) \in \mathbb{Z}_q \times \mathbb{Z}^m$ and has to prove

$$\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \tau\mathbf{G}\mathbf{v}_2 - \mathbf{D}\mathbf{m} = \mathbf{u} \bmod q,$$

where $\mathbf{v}_1 \in \mathbb{Z}^{m_1}$ and $\mathbf{v}_2 \in \mathbb{Z}^{m_2}$, with $\|\mathbf{v}_1\|_\infty \leq \sigma_1 \log_2 m_1$, $\|\mathbf{v}_2\|_\infty \leq \sigma \log_2 m_2$, $\tau \in \mathcal{T}$ and $\mathbf{m} \in \{0,1\}^{m_3}$. We define

$$\begin{cases} \alpha_1 = \lceil \sigma_1 \log_2 m_1 \rceil & k_{\alpha_1} = \lfloor \log_2 2\alpha_1 \rfloor + 1 & \mathbf{g}_{\alpha_1} = \left[\lfloor (2\alpha_1 + 2^{i-1})/2^i \rfloor\right]_{i \in [k_{\alpha_1}]} \\ \alpha = \lceil \sigma \log_2 m_2 \rceil & k_\alpha = \lfloor \log_2 2\alpha \rfloor + 1 & \mathbf{g}_\alpha = \left[\lfloor (2\alpha + 2^{i-1})/2^i \rfloor\right]_{i \in [k_\alpha]} \\ & k_{q'} = \lfloor \log_2 q' \rfloor + 1 & \mathbf{g}_{q'} = \left[\lfloor (q' + 2^{i-1})/2^i \rfloor\right]_{i \in [k_{q'}]} \end{cases}$$

Further, we define $\mathbf{a}_1 = \alpha_1 \mathbf{1}_{m_1}$ and $\mathbf{a} = \alpha \mathbf{1}_{m_2}$. Next, we set $\mathbf{G}_{\alpha_1} = \mathbf{I}_{m_1} \otimes \mathbf{g}_{\alpha_1}$, and $\mathbf{G}_\alpha = \mathbf{I}_{m_2} \otimes \mathbf{g}_\alpha$. We define $\mathbf{v}_1' = \mathbf{v}_1 + \mathbf{a}_1$, and $\mathbf{v}_2' = \mathbf{v}_2 + \mathbf{a}$. We then denote $\overline{\mathbf{v}_j}$ their respective binary decomposition along $\mathbf{g}_{\alpha_1}, \mathbf{g}_\alpha$, i.e., such that $\mathbf{G}_{\alpha_1}\overline{\mathbf{v}_1} = \mathbf{v}_1'$, and $\mathbf{G}_\alpha \overline{\mathbf{v}_2} = \mathbf{v}_2'$. We also denote by $\overline{\tau}$ the binary decomposition of τ along $\mathbf{g}_{q'}$ such that $\tau = \mathbf{g}_{q'}\overline{\tau}$. We need however to deal with the product term $\tau\mathbf{v}_2$. We use

[11] Choosing $\mathbf{g}_{\alpha_3}$ this way ensures that for any binary vector $\mathbf{x}$, $\mathbf{g}_{\alpha_3}\mathbf{x} \in [0, 2\alpha_3]$.

the same idea as for subset-sums from the framework of Yang et al. [YAZ+19]. For that, we define $\mathbf{u}_2 = \mathbf{G}\mathbf{v}_2 \in \mathbb{Z}^n$, and $\mathbf{u}_2' = \tau \mathbf{u}_2$. This gives an additional linear relation, but fewer decompositions. The prover now has to prove that

$$\begin{cases} \mathbf{A}\mathbf{G}_{\alpha_1}\overline{\mathbf{v}_1} - \mathbf{B}\mathbf{G}_{\alpha}\overline{\mathbf{v}_2} + \mathbf{u}_2' - \mathbf{D}\mathbf{m} = \mathbf{u} + \mathbf{A}\mathbf{a}_1 - \mathbf{B}\mathbf{a} \bmod q \\ \mathbf{G}\mathbf{G}_{\alpha}\overline{\mathbf{v}_2} - \mathbf{u}_2 = \mathbf{G}\mathbf{a} \bmod q \\ -\tau + \mathbf{g}_{q'}\overline{\tau} = 0 \bmod q \end{cases}$$

We thus define $\mathbf{x} = [\tau | \overline{\tau} | \overline{\mathbf{v}_1} | \overline{\mathbf{v}_2} | \mathbf{m} | \mathbf{u}_2 | \mathbf{u}_2'] \in \mathbb{Z}^{L_{\mathbf{x}}}$, where $L_{\mathbf{x}} = 1 + k_{q'} + m_1 k_{\alpha_1} + m_2 k_{\alpha} + m_3 + 2n$, as well as

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times k_{q'}} & \mathbf{A}\mathbf{G}_{\alpha_1} & -\mathbf{B}\mathbf{G}_{\alpha} & -\mathbf{D} & \mathbf{0}_{n\times n} & \mathbf{I}_n \\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times k_{q'}} & \mathbf{0}_{n\times m_1 k_{\alpha_1}} & \mathbf{G}\mathbf{G}_{\alpha} & \mathbf{0}_{n\times m_3} & -\mathbf{I}_n & \mathbf{0}_{n\times n} \\ -1 & \mathbf{g}_{q'} & \mathbf{0}_{n\times m_1 k_{\alpha_1}} & \mathbf{0}_{1\times m_2 k_{\alpha}} & \mathbf{0}_{1\times m_3} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} \end{bmatrix}$$

and $\mathbf{y} = [\mathbf{u} + \mathbf{A}\mathbf{a}_1 - \mathbf{B}\mathbf{a} | \mathbf{G}\mathbf{a}' | 0] \bmod q \in \mathbb{Z}_q^{2n+1}$. Finally, we define $\mathcal{M}_1 = \{(i,i,i); i \in [2, 1 + k_{q'} + m_1 k_{\alpha_1} + m_2 k_{\alpha} + m_3]\}$, which corresponds to the coefficients that need to be binary. We then need to add the relations $\mathbf{u}_2' = \tau \mathbf{u}_2$. For that, we define

$$\begin{aligned} \mathcal{M}_2 = \{ & (1 + k_{q'} + m_1 k_{\alpha_1} + m_2 k_{\alpha} + m_3 + n + i, 1, \\ & 1 + k_{q'} + m_1 k_{\alpha_1} + m_2 k_{\alpha} + m_3 + i); \;\; i \in [n]\}, \end{aligned}$$

and construct $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. The witness has length $L_{\mathbf{x}}$, and $\mathcal{M}$ is of size $L_{\mathcal{M}} = L_{\mathbf{x}} - n - 1$. Using the *fast mode* instead proves that $\mathbf{H}_1\mathbf{v}_1, \mathbf{H}_2\mathbf{v}_2$ have coefficients bounded by $\sigma_1\sqrt{m_1}\log_2\lambda$ and $\sigma\sqrt{m_2}\log_2\lambda$ respectively. It yields a witness of size $L_{\mathbf{x}} = 1 + k_{q'} + m_1 + m_2 + m_3 + 2n + k(\lfloor \log_2(2\sigma_1\sqrt{m_1}\log_2\lambda)\rfloor + \lfloor \log_2(2\sigma\sqrt{m_2}\log_2\lambda)\rfloor + 2)$, with $k = \lambda/\log_2(9/5)$, and $L_{\mathcal{M}} = L_{\mathbf{x}} - m_1 - m_2 - n - 1$.

Remark F.1. In the case where $q' = q$, the tag does not need to be decomposed in binary form. However, when the proof system is run only a few number of times, we need to drastically increase the size of challenges to reach a negligible soundness error. For example, to obtain a negligible soundness error in one iteration, one needs to take challenges of size $p = 2^{\lambda}$. Because the SIS bound for the proof system is $\beta_{\infty} = \mathsf{poly}(\lambda) \cdot p^2$, one must take q to be polynomially larger than p^2. In Algorithm 3.1, choosing $q' = q$ then leads to a tag space $\mathcal{T}$ of size at least $\mathsf{poly}(\lambda)2^{2\lambda}$. As a result, the proof of Lemma 3.1 incurs an exponential reduction loss of $1/|\mathcal{T}| = 2^{-2\lambda}/\mathsf{poly}(\lambda)$. To circumvent this limitation, one can choose $q' = \mathsf{poly}(\lambda) \ll q$ to make the reduction loss acceptable. It implies that the signature verification must ensure that $\tau < q'$, which we consider when proving possession of a message-signature pair.

F.3 Instantiating [LLM+16] with $\mathcal{R}^*$

The original construction by Libert et al. [LLM+16] uses the binary decomposition of the commitment $\mathbf{c}$ instead of using the commitment itself. It additionally bases itself on the Boyen signature scheme, and involves an extra matrix $\mathbf{D} \in \mathbb{Z}_q^{n \times 2nk}$, where $k = \lceil \log_2 q \rceil$. For a fair comparison, we detail here how to use the framework from [YAZ+19] for the construction of [LLM+16]. For this section only, we set the parameters differently according to [LLM+16]. We thus have $m = 2nk$, $\sigma_1 = \sigma\sqrt{1 + 8(N+1)^2 m^3}$. Also, prior to being signed, the message blocks are encoded using $b \mapsto (1-b, b)$. This means that although the relevant message information is of mN bits, it is treated as a message of $2mN$ bits. To be thorough, one would need to prove that the message is properly encoded in addition to proving that the message is binary. This can be done by proving the additional relation $(\mathbf{I}_{mN} \otimes [1\ 1])\mathbf{m} = \mathbf{1}_{mN}$ which proves that the consecutive bits $b, 1-b$ indeed sum to 1. Since the relation is proven modulo q, one must make sure that the coefficients are $\mathbf{m}$ are also proven binary. For simplicity, we do not take this into account in the estimations of Table G.1. The matrices $\mathbf{A}_i$ are uniform in $\mathbb{Z}_q^{n \times m}$, but the commitment key matrices $\mathbf{D}_i$ are uniform in $\mathbb{Z}_q^{2n \times 2m}$. We define $\mathbf{H} = \mathbf{I}_{2n} \otimes [2^0 \ldots 2^{k-1}]$. Since the binary decomposition operator is non-linear, the verification equation has to be splitted into two equations as follows.

$$\begin{cases} \mathbf{A}\mathbf{v}_1 + \mathbf{A}_0\mathbf{v}_2 + \sum_{i\in[\ell]} \mathbf{A}_i(\tau[i]\mathbf{v}_2) - \mathbf{D}\mathbf{w} = \mathbf{u} \bmod q, \\ \mathbf{H}\mathbf{w} = \mathbf{D}_0\mathbf{r} - \sum_{i\in[N]} \mathbf{D}_i\mathbf{m}_i \bmod q, \end{cases}$$

with $\|\mathbf{v}_1\|_\infty, \|\mathbf{v}_2\|_\infty \leq \sigma \log_2 m$, $\|\mathbf{r}\|_\infty \leq \sigma_1 \log_2 2m$ as well as $\tau \in \{0,1\}^\ell, \mathbf{m} \in \{0,1\}^{2mN}$, and $\mathbf{w} \in \{0,1\}^{2nk}$. We define $\alpha, \alpha_1, k_\alpha, k_{\alpha_1}, \mathbf{g}_\alpha, \mathbf{g}_{\alpha_1}$ in a similar way as Section F. We then define $\mathbf{a} = \alpha\mathbf{1}_m$, $\mathbf{a}_1 = \alpha_1\mathbf{1}_{2m}$ and set $\mathbf{G}_\alpha = \mathbf{I}_m \otimes \mathbf{g}_\alpha$ and $\mathbf{G}_{\alpha_1} = \mathbf{I}_{2m} \otimes \mathbf{g}_{\alpha_1}$. Then, we define $\mathbf{u}_i = \mathbf{A}_i\mathbf{v}_2 \in \mathbb{Z}^n$, as well as $\mathbf{u}'_i = \tau[i]\mathbf{u}_i$ constituting 2ℓ vectors[12] of $\mathbb{Z}^n$. The verification equations thus become

$$\begin{cases} \mathbf{A}\mathbf{G}_\alpha\overline{\mathbf{v}_1} + \mathbf{A}_0\mathbf{G}_\alpha\overline{\mathbf{v}_2} + \sum_{i\in[\ell]} \mathbf{u}'_i - \mathbf{D}\mathbf{w} = \mathbf{u} + (\mathbf{A} + \mathbf{A}_0)\mathbf{a} \bmod q, \\ \mathbf{D}_0\mathbf{G}_{\alpha_1}\overline{\mathbf{r}} + \sum_{i\in[N]} \mathbf{D}_i\mathbf{m}_i - \mathbf{H}\mathbf{w} = \mathbf{D}_0\mathbf{a}_1 \bmod q, \\ \mathbf{A}_i\mathbf{G}_\alpha\overline{\mathbf{v}_2} - \mathbf{u}_i = \mathbf{A}_i\mathbf{a} \text{ for all } i \in [\ell], \end{cases}$$

We thus define $\mathbf{x} = [\tau|\overline{\mathbf{v}_1}|\overline{\mathbf{v}_2}|\mathbf{u}_1|\ldots|\mathbf{u}_\ell|\mathbf{u}'_1|\ldots|\mathbf{u}'_\ell|\mathbf{w}|\overline{\mathbf{r}}|\mathbf{m}_1|\ldots|\mathbf{m}_N] \in \mathbb{Z}^{L_\mathbf{x}}$, where $L_\mathbf{x} = \ell + 2mk_\alpha + 2\ell n + m + 2mk_{\alpha_1} + 2mN$. We then define

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{A}\mathbf{G}_\alpha & \mathbf{A}_0\mathbf{G}_\alpha & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_n & \cdots & \mathbf{I}_n & -\mathbf{D} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ & & & & & & & & & -\mathbf{H} & \mathbf{D}_0\mathbf{G}_{\alpha_1} & \mathbf{D}_1 & \cdots & \mathbf{D}_N \\ & & \mathbf{A}_1\mathbf{G}_\alpha & -\mathbf{I}_n & & & & & & & & & & \\ & & \vdots & & \ddots & & & & & & & & & \\ & & \mathbf{A}_\ell\mathbf{G}_\alpha & & & -\mathbf{I}_n & & & & & & & & \end{bmatrix},$$

[12] Each bit of the tag not only represents a witness in itself but also entails two full intermediate witnesses $\mathbf{u}_i, \mathbf{u}'_i$ due to $\tau[i]\mathbf{v}_2$. It leads to a much larger witness vector, which is a source of inefficiencies.

and

$$\mathbf{y} = \begin{bmatrix} \mathbf{u} + (\mathbf{A} + \mathbf{A}_0)\mathbf{a} \\ \mathbf{D}_0\mathbf{a}_1 \\ \mathbf{A}_1\mathbf{a} \\ \vdots \\ \mathbf{A}_\ell\mathbf{a} \end{bmatrix}.$$

Finally, we define $\mathcal{M}_1 = \{(i,i,i); i \in [\ell + 2mk_\alpha] \cup [\ell + 2mk_\alpha + 2\ell n + 1, L]\}$, which corresponds to having all the coefficients of $\mathbf{x}$ to be binary, except for the $\mathbf{u}_i, \mathbf{u}'_i$. We then need to add the relations $\mathbf{u}'_i = \tau[i]\mathbf{u}_i$ for $i \in [\ell]$. For that, we define

$$\mathcal{M}_2 = \{(\ell + 2mk_\alpha + \ell n + n(i-1) + j, i, \ell + 2mk_\alpha + n(i-1) + j); (i,j) \in [\ell] \times [n]\},$$

and construct $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. The witness has length $L_\mathbf{x}$, and $\mathcal{M}$ is of size $L_\mathbf{x} - \ell n$ as well. The fast mode of Section 4.2 reduces the witness and relation set sizes to

$$\begin{cases} L_\mathbf{x} = \ell(2n+1) + m(5+2N) + k(2\lfloor \log_2(2\sigma\sqrt{m}\log_2\lambda)\rfloor \\ \qquad\qquad\qquad\qquad\qquad\qquad + \lfloor \log_2(2\sigma_1\sqrt{2m}\log_2\lambda)\rfloor + 3) \\ L_\mathcal{M} = L_\mathbf{x} - \ell n - 4m, \end{cases}$$

where $k = \lambda / \log_2(9/5)$ according to Lemma 4.1.

G Parameters and Efficiency

In this section, we instantiate the two versions of our signature scheme with concrete parameters in order to reach $\lambda = 128$ bits of quantum security. All the concrete hardness estimates for the SIS, LWE, M-SIS, M-LWE problems are done with the Core-SVP methodology, using the BKZ cost model with sieving SVP oracle. In this model, the classical security is given by $\lambda_c = 0.292b$ [BDGL16] and the quantum security by $\lambda_q = 0.265b$ [Laa15], where b is the BKZ blocksize. We explain our choice of parameters for both the standard and module version by encompassing the zero-knowledge arguments of message-signature possession. We however note that for a standard use of the signature schemes, one could choose different parameters. We choose to instantiate it for $Q = 2^{30}$ signature queries, representing the number of signature issuance. We believe this choice is reasonable for most applications.

G.1 Instantiating the Standard Signature

We provide in Table G.2 an example parameter set along with the size of the keys, signature, and proof of possession for the signature of Section 3. It makes use of the zero-knowledge framework of [YAZ+19] improved with the enhanced fast mode from Section 4.2 and the optimizations of Appendix E (except the compacted commitments, as explained below) that we have introduced.

As explained in Remark F.1, in order to have as few iterations of the proof system as possible, we need to choose large enough challenges, which in turns require to take a sufficiently large modulus. We then start by choosing the number of iterations N and the challenge size p, which imply we must take $q \geq \mathsf{poly}(\lambda) \cdot p^2$. To avoid an exponential reduction loss, we set $q' \approx Q$. We then fix n so that when the other parameters are set using Algorithm 3.1, we obtain a quantum security of λ. Since the proofs of Lemma 3.1 and 3.2 both have a reduction loss between the advantage of a signature forger ($\delta = 2^{-\lambda}$) and the advantage against SIS (Adv[$\mathcal{B}$]) which can be substantial, we need to take it into account. More precisely, we compute the required SIS security $\lambda_{\mathrm{I}}, \lambda_{\mathrm{II}}$ so that the SIS problem stays hard even with the relations of Lemma 3.1 and 3.2. For our parameter, we need $\lambda_{\mathrm{I}} = 189$ and $\lambda_{\mathrm{II}} = 181$ for the respective SIS problems which only slightly differ by their bounds. Hence, we must reach for a root Hermite factor of $\delta_0 = 1.0026$. We also account for key recovery attacks, consisting of recovering $\mathbf{R}$ from $\mathbf{A}, \mathbf{B}$. This attack is however much more costly than forgeries as $\mathbf{R}$ is *statistically* hidden in $(\mathbf{A}, \mathbf{B})$ by the leftover hash lemma. We then set the other parameters of the zero-knowledge argument as described in our optimized framework in Appendix E and taking ℓ_1, ℓ_2 to reach 128 bits of quantum security for the HNF-SIS and HNF-LWE problems. The security estimates of HNF-LWE are performed using the estimator of Albrecht et al. [APS15]. We note that although we take the secret and error ternary from distribution ψ_1, we are never in the regime of polynomial algebraic attacks [AG11]. Such attacks for ternary error would require roughly ℓ_2^3 samples. In our cases, we have $\ell_1 + \max(L_{\mathbf{x}}, L_{\mathcal{M}}) \ll \ell_2^2$.

We also instantiate the scheme of [LLM$^+$16]. For a fair comparison, we aim for the same security and make use of the same improvements of the zero-knowledge argument. The relation of [LLM$^+$16] is instantiated in the framework of [YAZ$^+$19] in Appendix F.3.

For both our scheme and the one from [LLM$^+$16], the ZKAoK are instantiated to be run twice, and thus do not include the compacted commitments discussed in Appendix E. Table 1.1 shows the construction of [LLM$^+$16] leads to intractable parameters and key sizes. We note that one could reduce the value of q at the expense of increasing the number of proof iterations to achieve negligible soundness. However, not only does this approach still leads to intractable key sizes, but it also yields substantially larger proofs. Our results also summarized in Table 1.1 shows the feasibility of signature with efficient protocols based on lattice assumptions, as we gain several orders of magnitude in the size of key materials and proof size, while maintaining the same security. The complete parameter sets used to obtained these results can be found in Tables G.1 and G.2.

Remark G.1. We recall that, although the fast mode reduces the size of the witness vector, it also introduces a soundness gap, which is the object of Lemma 4.1. As a result, the bounds on $\mathbf{v}_1^*, \mathbf{v}_2^*$ used in Lemma 3.1 and 3.2 are larger as discussed in Remark 4.1. We thus take this increase of the SIS bounds into account when estimating the SIS security, which entails an increase of the dimension n.

G.2 Instantiating the Module Signature

We now rely on the framework of [LNP22] for the zero-knowledge argument. The module construction no longer suffers from the requirement of a large modulus. Indeed, in the module case, we can choose an exponentially large challenge space while keeping the size of the challenges constant. The same thing occurs for our tag space. Before, we needed to take $q \geq q'$ where q' was both the bound on the tags and the size of the tag space. In the module case, we can take binary tags while adjusting the value of w in order to have a sufficiently large tag space, i.e., $|\mathcal{T}_w| \geq Q$. Additionally, because the modulus of the signature q is different from the modulus of the proof system $q_\pi = q_1 q$, we can first adjust the parameters of our signature before setting the parameters of the proof system. We proceed as in the previous section, accounting for the reduction loss of Lemma 3.4 and 3.5. To choose the parameters of the proof system, we proceed as prescribed in [LNP22, Sec. 6.1], with the challenge space of [LNP22, Fig. 3]. For simplicity, we choose parameters close to those provided in their group signature instantiation. We give the detailed parameter set in Table G.3 with security and efficiency estimates. To avoid collision between our notations and the proof system parameters, we specify the notations used in [LNP22] in the description column.

This construction based on structured lattices leads to drastic efficiency gains in both key and proof sizes as summarized in Table 1.1, which further reinforce the concrete feasibility of efficient privacy-enhancing post-quantum signatures. In particular, it shows that a proof of knowledge of a signature issued on a committed (secret value), one of the main building blocks of privacy-preserving primitives, can represent less than 700 KB, which is a considerable improvement over [LLM+16] and may have many applications.

Instantiating the Anonymous Credentials. The anonymous credentials follows exactly the same process as for the module signature. The only difference is that we in addition require the hardness of M-ISIS$_{d,2d,q,\sqrt{2nd}}$. The parameters of our signature already overshot the hardness of the latter. We give an example for a vector of 10 attributes of 128 bits each, and the proof size corresponds to the non-interactive transcript size when $|\mathcal{I}| = 4$ attributes are revealed. The message $\widetilde{\mathbf{m}}$ thus has dimension $m_3 = 2d + 10$. All the parameters and efficiency estimates of the anonymous credentials are given in Table G.4. We achieve satisfactory sizes of less than 750 KB. Improving our SEP scheme would directly result in similar improvements to the anonymous credentials system.

G.3 Parameter Sets

Parameters	Description	Exact Proof	Fast Mode
Signature			
λ	Security parameter	128	128
n	SIS dimension	1650	2550
q	Modulus	$2^{155} - 31$	$2^{155} - 31$
ℓ	Tag bit-size ($\lambda + 2\log_2 Q$)	188	188
m	Trapdoor dimension	511500	790500
N_{msg}	Number of message blocks	1	1
σ	Pre-image sampling width	24324	32014
σ_1	Commitment randomness width	50335037584951	127285811917979
$\lambda_{\mathrm{I}}/\lambda_{\mathrm{I}}^*$	Required/Reached SIS security (I)	166/167	166/167
$\lambda_{\mathrm{II}}/\lambda_{\mathrm{II}}^*$	Required/Reached SIS security (II)	158/163	158/170
$\lambda_{\mathrm{III}}/\lambda_{\mathrm{III}}^*$	Required/Reached SIS security (III)	128/506	128/629
$\lvert\mathsf{pk}\rvert$	Public key size (MB)	$2963 \cdot 10^3$ MB	$7076 \cdot 10^3$ MB
$\lvert\mathsf{sk}\rvert$	Secret key size (MB)	$1559 \cdot 10^1$ MB	$3725 \cdot 10^1$ MB
$\lvert\mathsf{sig}\rvert$	Signature size (KB)	8617 **KB**	13895 **KB**
$\lvert\mathsf{pp}\rvert$	Public parameters size (MB)	$1640 \cdot 10^1$ MB	$3917 \cdot 10^1$ MB
Proof			
ℓ_1	HNF-SIS dimension	8350	8000
ℓ_2	HNF-LWE dimension	7900	7900
p	Size of challenges	$2^{\lambda/2}$	$2^{\lambda/2}$
N	Number of proof iterations	2	2
M	Rejection sampling repetition rate	27	27
$L_{\mathbf{x}}$	Witness length	74788088	6510457
$L_{\mathcal{M}}$	Relation set length	74477888	2869057
δ_s	Soundness error	$2^{-\lambda}$	$2^{-\lambda}$
$\lambda_{\mathrm{SIS},\pi}^*$	Reached HNF-SIS security	128	129
$\lambda_{\mathrm{LWE},\pi}^*$	Reached HNF-LWE security	130	130
$\lvert\pi\rvert$	Proof size (KB)	10198709 **KB**	671581 **KB**

Table G.1. Selected parameters, security and efficiency estimates of the signature scheme of [LLM+16].

Parameters	Description	Exact Proof	Fast Mode
Signature			
λ	Security parameter	128	128
n	SIS dimension	495	795
q	Modulus	$2^{155} - 31$	$2^{155} - 31$
q'	Tag bound	2^{31}	2^{31}
m_1	First trapdoor dimension	48732	78070
m_2	Second trapdoor dimension	76725	123225
m_3	Message bit-size	128	128
t	Spectral norm slack	7.5	7.5
σ	Pre-image sampling width	6015.41	7595.30
σ_1	$\sqrt{\sigma^2 + \sigma_2^2}$	6015.42	7595.31
σ_2	Commitment randomness width	12.73	12.73
$\lambda_{\mathrm{I}}/\lambda_{\mathrm{I}}^*$	Required/Reached SIS security (I)	158/190	158/189
$\lambda_{\mathrm{II}}/\lambda_{\mathrm{II}}^*$	Required/Reached SIS security (II)	182/190	182/189
$\lvert\mathsf{pk}\rvert$	Public key size (MB)	1148 MB	2956 MB
$\lvert\mathsf{sk}\rvert$	Secret Key size (MB)	892 MB	2293 MB
$\lvert\mathsf{sig}\rvert$	Signature size (KB)	261 **KB**	418 **KB**
$\lvert\mathsf{pp}\rvert$	Public parameters size (MB)	1.2 MB	1.9 MB
Proof			
ℓ_1	HNF-SIS dimension	7850	7500
ℓ_2	HNF-LWE dimension	7850	7850
p	Size of challenges	$2^{\lambda/2}$	$2^{\lambda/2}$
N	Number of proof iterations	2	2
M	Rejection sampling repetition rate	27	27
$L_{\mathbf{x}}$	Witness length	2259407	210777
$L_{\mathcal{M}}$	Relation set length	2258911	8686
δ_s	Soundness error	$2^{-\lambda}$	$2^{-\lambda}$
$\lambda_{\mathrm{SIS},\pi}^*$	Reached HNF-SIS security	128	129
$\lambda_{\mathrm{LWE},\pi}^*$	Reached HNF-LWE security	129	129
$\lvert\pi\rvert$	Proof size (KB)	306367 **KB**	17662 **KB**

Table G.2. Selected parameters, security and efficiency estimates of the signature scheme of Section 3.

Parameters	Description	Value
Signature		
λ	Security parameter	128
n	Ring degree	128
d	M-SIS module rank	10
q	Modulus	$2^{47} - 279$
k	Number of splitting factors	4
w	Tag norm bound	6
$\binom{n}{w}$	Size of tag space	$\approx 2^{32.3}$
κ	Gadget matrix term	$\lceil \log_2 q \rceil$
m_1	First trapdoor rank	620
m_2	Second trapdoor rank	470
m_3	Number of message polynomials	1
t	Spectral norm slack	7.5
σ	Pre-image sampling width	5379
σ_1	$\sqrt{\sigma^2 + \sigma_2^2}$	5935
σ_2	Commitment randomness width	2510
$\lambda_{\mathrm{I}}/\lambda_{\mathrm{I}}^*$	Required/Reached M-SIS security (I)	161/192
$\lambda_{\mathrm{II}}/\lambda_{\mathrm{II}}^*$	Required/Reached M-SIS security (II)	182/184
$\lvert\mathsf{pk}\rvert$	Public key size (MB)	7.82 MB
$\lvert\mathsf{sk}\rvert$	Secret Key size (MB)	8.89 MB
$\lvert\mathsf{sig}\rvert$	Signature size (KB)	273 **KB**
$\lvert\mathsf{pp}\rvert$	Public parameters size (MB)	0.007 MB
Proof		
d'	Height of commitment matrices $\mathbf{A}_1, \mathbf{A}_2$ (n)	17
q_1	Slack Modulus (q_1)	$2^{28} - 119$
q_π	Proof modulus (q)	$\approx 2^{75}$
-	Bound on challenges (κ)	2
$\lvert\mathcal{C}\rvert$	Size of challenge space ($\lvert\mathcal{C}\rvert$)	$\approx 2^{147}$
σ_{-1}	Proof automorphism (σ)	σ_{-1}
η	Second bound on challenges (η)	72
ν	Randomness $\mathbf{s}_2$ bound (ν)	1
-	Number of garbage terms (λ)	5
-	Length of $\mathbf{s}_1$ (m_1)	1092
-	Length of $\mathbf{m}$ (ℓ)	0
-	Length of $\mathbf{s}_2$ (m_2)	41
γ_1	Rejection sampling constant for $c\mathbf{s}_1$ (γ_1)	5
γ_2	Rejection sampling constant for $c\mathbf{s}_2$ (γ_2)	3
$\gamma^{(e)}$	Rejection sampling constant for exact ARP ($\gamma^{(e)}$)	2
δ_s	Soundness error	$\approx 2^{-140}$
$\lambda^*_{\text{M-SIS},\pi}$	Reached M-SIS security	212
$\lambda^*_{\text{M-LWE},\pi}$	Reached ext-M-LWE security	141
$\lvert\pi\rvert$	Proof size (KB)	639 **KB**

Table G.3. Selected parameters, security and efficiency estimates of the signature scheme of Section 3.3.

Parameters	Description	Value
Signature		
λ	Security parameter	128
n	Ring degree	128
d	M-SIS module rank	12
q	Modulus	$2^{44} - 119$
k	Number of splitting factors	4
w	Tag norm bound	6
$\binom{n}{w}$	Size of tag space	$\approx 2^{32.3}$
κ	Gadget matrix term	$\lceil \log_2 q \rceil$
m_1	First trapdoor rank	657
m_2	Second trapdoor rank	528
m_s	Dimension of user secret key	24
m'_3	Number of attribute polynomials	10
t	Spectral norm slack	7.5
σ	Pre-image sampling width	5609
σ_1	$\sqrt{\sigma^2 + \sigma_2^2}$	26587
σ_2	Commitment randomness width	25989
$\lambda_{\text{I}}/\lambda_{\text{I}}^*$	Required/Reached M-SIS security (I)	161/191
$\lambda_{\text{II}}/\lambda_{\text{II}}^*$	Required/Reached M-SIS security (II)	182/183
$\lvert\mathsf{pk}\rvert$	Public key size (MB)	9.56 MB
$\lvert\mathsf{sk}\rvert$	Secret Key size (MB)	10.59 MB
$\lvert\mathsf{sig}\rvert$	Signature size (KB)	317 **KB**
$\lvert\mathsf{pp}\rvert$	Public parameters size (MB)	0.27 MB
Proof		
d'	Height of commitment matrices $\mathbf{A}_1, \mathbf{A}_2$ (n)	17
q_1	Slack Modulus (q_1)	$2^{28} - 119$
q_π	Proof modulus (q)	$\approx 2^{72}$
-	Bound on challenges (κ)	2
$\lvert\mathcal{C}\rvert$	Size of challenge space $(\lvert\mathcal{C}\rvert)$	$\approx 2^{147}$
σ_{-1}	Proof automorphism (σ)	σ_{-1}
η	Second bound on challenges (η)	72
ν	Randomness $\mathbf{s}_2$ bound (ν)	1
-	Number of garbage terms (λ)	5
$\lvert\mathcal{I}\rvert$	Number of disclosed attributes	4
-	Length of $\mathbf{s}_1$ (m_1)	1216
-	Length of $\mathbf{m}$ (ℓ)	0
-	Length of $\mathbf{s}_2$ (m_2)	41
γ_1	Rejection sampling constant for $c\mathbf{s}_1$ (γ_1)	5
γ_2	Rejection sampling constant for $c\mathbf{s}_2$ (γ_2)	3
$\gamma^{(e)}$	Rejection sampling constant for exact ARP $(\gamma^{(e)})$	2
δ_s	Soundness error	$\approx 2^{-140}$
$\lambda^*_{\text{M-SIS},\pi}$	Reached M-SIS security	180
$\lambda^*_{\text{M-LWE},\pi}$	Reached ext-M-LWE security	149
$\lvert\pi\rvert$	Proof size (KB)	724 **KB**

Table G.4. Selected parameters, security and efficiency estimates for the anonymous credentials of Section 5.3.

H A Stateless Variant

We now describe how to obtain a stateless version of our signature scheme of Section 3. We describe it for the module version at a high level only. To obtain a stateless variant, one need to find a way to generate tags without encountering collisions to only emit one signature per tag, and without enduring an exponential loss in the security proof due to guesses. The first constraint imposes an exponentially large tag space, which at first glance seems incompatible with the second constraint. To meet the second constraint, we use the prefix guessing method used in [LLM+16] which we refine to avoid a blowup in the key size and subsequent zero-knowledge proofs. At a high level, when guessing a tag $\tau \in \{0,1\}^\ell$ in [LLM+16], the authors guess the longest prefix among the emitted tags $\tau^{(i)}$ that will match with the prefix of the tag τ^* used in the type I forgery. In our case, our tag is a single element of S_{bin} which voids this method as each bit would need to be multiplied to a different matrix $\mathbf{A}_i$ to cancel the gagdet $\mathbf{G}$ when $\tau = \tau^*$ in order to obtain a M-SIS solution.

To mimic the prefix guessing, we remove the Hamming weight condition on the tag $\tau \in S_{bin}$ and instead decompose it into k parts, say $k = 4$, as follows. Sample $\tau_1, \ldots, \tau_4$ uniformly at random in $\sum_{i=0}^{n/4-1} \{0,1\} \cdot X^i$. To construct the trapdoor matrix $\mathbf{A}_\tau$, compute $\mathbf{A}_\tau = [\mathbf{A} | \sum_{i\in[4]} \tau_i \mathbf{B}_i - \mathbf{B}]$ where $\mathbf{B}_i = X^{(i-1)n/4}\mathbf{G} - \mathbf{A}\mathbf{R}_i \bmod qR$. The $\mathbf{B}_i$ are added to the public key, and the $\mathbf{R}_i$ to the secret key. We then have

$$\mathbf{A}_\tau = [\mathbf{A} | \tau\mathbf{G} - \mathbf{A}(\mathbf{R} + \sum_{i\in[4]} \tau_i \mathbf{R}_i)].$$

Since $\tau = \tau_1 + \tau_2 X^{n/4} + \tau_3 X^{n/2} + \tau_4 X^{3n/4} \in S_{bin} \subset R_q^\times$, we can indeed sample preimages using the modified trapdoor $\mathbf{R}_\tau = \mathbf{R} + \sum_i \tau_i \mathbf{R}_i$. This has the effect of requiring a slightly larger Gaussian width as a result because we must replace $\|\mathbf{R}\|_2$ by

$$\begin{aligned} \|\mathbf{R}_\tau\|_2 &\leq \|\mathbf{R}\|_2 + \sum_{i\in[4]} \|\tau_i\|_1 \|\mathbf{R}_i\|_2 \\ &\leq (1+n) \cdot (\sqrt{nm_1} + \sqrt{nm_2} + t). \end{aligned}$$

It thus entails a mild increase in σ, which only affects the width of $\mathbf{v}_2$ in the signature as the width σ_1 for $\mathbf{v}_1$ smoothes out this increase by requiring $\sigma_1 \geq \|\mathbf{U}\mathbf{m}\|_2$ which is generally larger for a big enough m_3. Regardless, the factor n, typically 256, only adds $\log_2 n$ bits to each coefficient.

With this modification, we can use the prefix guessing method. Concretely, as in [LLM+16], we simply make a guess on the length ℓ^+ of the longest prefix common to the forgery tag and the ones from the signature queries. In our case, this guess is correct with probability $1/k$. We thus only have to guess the value of $\tau^*_{\ell^+ + 1}$ and construct the key material accordingly. We will thus hide the guessed prefix in the matrix $\mathbf{B}$, and generate the $\mathbf{B}_i$ simply as $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i$ for the indices i beyond the prefix length. With this method, we only have a security loss of $k2^{n/k}$ while having an exponentially large tag space of size 2^n.