Practical Post-Quantum Signatures for Privacy

Sven Argo[1], Tim Güneysu[1,2], Corentin Jeudy[3,4], Georg Land[1],
Adeline Roux-Langlois[5], and Olivier Sanders[3]

sven.argo@rub.de, tim.gueneysu@rub.de, corentin.jeudy@orange.com,
mail@georg.land, adeline.roux-langlois@cnrs.fr, olivier.sanders@orange.com

[1] Ruhr University Bochum, Horst Görtz Institute for IT-Security, Bochum, Germany
[2] DFKI GmbH, Cyber-Physical Systems, Bremen, Germany
[3] Orange Labs, Applied Crypto Group, Cesson-Sévigné, France
[4] Univ Rennes, CNRS, IRISA, Rennes, France
[5] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract. The transition to post-quantum cryptography has been an enormous challenge and effort for cryptographers over the last decade, with impressive results such as the future NIST standards. However, the latter has so far only considered central cryptographic mechanisms (signatures or KEM) and not more advanced ones, e.g., targeting privacy-preserving applications. Of particular interest is the family of solutions called blind signatures, group signatures and anonymous credentials, for which standards already exist, and which are deployed in billions of devices. Such a family does not have, at this stage, an efficient post-quantum counterpart although very recent works improved this state of affairs by offering two different alternatives: either one gets a system with rather large elements but a security proved under standard assumptions or one gets a more efficient system at the cost of ad-hoc interactive assumptions or weaker security models. Moreover, all these works have only considered size complexity without implementing the quite complex building blocks their systems are composed of. In other words, the practicality of such systems is still very hard to assess, which is a problem if one envisions a post-quantum transition for the corresponding systems/standards.

In this work, we propose a construction of so-called signature with efficient protocols (SEP), which is the core of such privacy-preserving solutions. By revisiting the approach by Jeudy et al. (Crypto 2023) we manage to get the best of the two alternatives mentioned above, namely short sizes with no compromise on security. To demonstrate this, we plug our SEP in an anonymous credential system, achieving credentials of less than 80 KB. In parallel, we fully implemented our system, and in particular the complex zero-knowledge framework of Lyubashevsky et al. (Crypto'22), which has, to our knowledge, not be done so far. Our work thus not only improves the state-of-the-art on privacy-preserving solutions, but also significantly improves the understanding of efficiency and implications for deployment in real-world systems.

An extended abstract of this work appeared at CCS 2024 which can be accessed at https://doi.org/10.1145/3658644.3670297. This is the full version.

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

1 Introduction

Digital signatures have become pervasive in electronic systems, ensuring authentication at very moderate cost. We use them on a daily basis, to secure web browsing, digital payments and ID documents, e.g., passports. This cryptographic primitive enables an *issuer* to authenticate some set of data $\{m_i\}$ by generating a signature sig that can be verified with the sole knowledge of the issuer's public key. In practice, this can be done with a remarkable efficiency. From the security standpoint, the situation also seems entirely satisfactory. Current standards like ECDSA are more than 25 years old and have so far withstood all cryptanalytic attempts. It might then seem that this area of cryptography is set to evolve quietly, without significant hitches, until transition to post-quantum cryptography becomes legally mandatory.

This seemingly ideal situation must be considered with caution as standard digital signatures have inherent limitations that are undesirable in many use-cases. One of these limitations is that the verification of the signature sig requires the knowledge of the full set $\{m_i\}$, even if one is only interested in checking the authenticity of a single element of this set. In the context of digital identity, this concretely means that a user must reveal all his attributes, e.g., name, address, date of birth, etc, to prove authenticity of only one of them. Another limitation is the traceability enabled by the signature. Each presentation of the signature involves sending the same value sig which can be used to trace its owner.

The topical example of age control to access adult-only websites epitomizes these problems. The current debates in France[6] or United Kingdom[7] show the same divide between two groups. One group is obviously unhappy with the current declarative approach, where the user certifies being old enough to access the website, and thus calls for stronger forms of authentication. Digital certificates could easily address this problem but the other group points out the obvious privacy issues resulting from the limitations mentioned above. Actually, unnecessarily providing sensitive information to a website is likely to lead to severe security issues that go well beyond mere privacy concerns: phishing, impersonation, etc.

Fortunately, we are not stuck with this endless debate on law enforcement versus personal liberties. For decades, cryptographers have indeed worked to devise privacy-preserving authentication mechanisms that could reconcile these two sides. According to the use-cases they address, these mechanisms are called blind signatures [Cha82], group signatures [CvH91], DAA [BCC04], EPID [BL07], anonymous credentials [FHS19], etc. but they all share the same fundamental security principle: limiting information disclosure to what is strictly functionally necessary. Far from being mere theoretical contributions, these mechanisms can

[6] CNIL recommendations for online age verification and user privacy

[7] United Kingdom safety bill strengthening age verification

be implemented very efficiently [PS16,CDL16,San21] leading to a small overhead compared to a non-private version built upon standard digital signatures. Some of them have been included in standards [ISO13a,ISO13b] and even embedded in billions of devices [TCG15,Int16]. Very recently, they have been advocated[8] by the GSMA (an organization gathering most industrial actors of the telecommunication ecosystem) for implementing the future European Digital Identity Wallet[9]. Interestingly, this GSMA document depicts privacy as a "positive differentiator", thus contrasting with the usual perception of privacy which was so far seen as a legal constraint. If it reflects an evolution of the industrial position on this topic, then we could see more applications of those privacy-preserving mechanisms in a near future.

Incidentally, this evolution towards more private systems coincides with a post-quantum transition that is urged by most security agencies across the world. A natural question before adopting a group signature or an anonymous credential scheme is then whether there exists an efficient post-quantum variant that could take over when the quantum threat will become more tangible. Clearly, the situation is not as positive as the one of classical[10] cryptography. Although some post-quantum variants of the primitives above have been proposed, e.g., [dPLS18,CKLL19,BEF19], we note that they suffered from quite large sizes that are likely to be incompatible with industrial constraints.

To understand the challenges faced when designing post-quantum versions of these privacy-preserving authentication mechanisms, it is necessary to recall how they work at a very high level. Such mechanisms do not fundamentally change the authentication paradigm as they all rely on a central issuer that generates signatures on the users' data to authenticate them. The difference with the standard approach lies in the way this signature is obtained and then presented by the users. In some situations, the user may indeed need to obtain a signature on some hidden data. In some other situations, the user may have to use the received signature to authenticate some of his personal data while hiding the signature and the other signed data. All these situations call for zero-knowledge (ZK) proofs [GMR85] that are indeed designed to prove statements while hiding the corresponding witness. By carefully crafting the ZK proof, one can ensure a minimal leakage and thus achieve the privacy properties claimed by the privacy-preserving mechanisms. We note that this approach does not necessarily exclude standard digital signature schemes as one can always build dedicated ZK proofs to manage the situations described above. However, the resulting system is likely to be totally impractical.

To tackle the efficiency problem which is also in focus of this work, it is necessary to approach it from a very different perspective. Instead of starting from a standard digital signature scheme and then trying to adapt ZK proofs to it, it is better to design from scratch a signature scheme that will smoothly interact with ZK proofs. This is the approach successfully adopted by classical

[8] GSMA Official Response: eIDAS 2.0 and Privacy

[9] European Digital Identity Wallet Architecture and Reference Framework

[10] By "classical", we mean vulnerable to quantum computing.

cryptography which led to the design of several so-called "signatures with efficient protocols" (SEP) [CL04,BB08,PS16] and thereby to the remarkable efficiency or privacy-preserving mechanisms using them.

Concretely, a SEP is a signature scheme that comes with two main protocols: one for issuing signatures on committed messages, and one for efficiently proving knowledge of a signature. Thanks to these properties, it perfectly fits most of the generic frameworks used to build privacy-preserving primitives. For example, in the blind signature framework by Fischlin [Fis06], the signer first signs a committed message m and then sends the resulting signature S to the user. The latter derives a blind signature by encrypting S and producing a zero-knowledge proof that the encrypted S is indeed a valid signature on a commitment of m. Generating S with a SEP scheme is thus particularly relevant as it inherently supports such features. Similarly, in the group signature framework by Bellare, Shi and Zhang [BSZ05], a signature must contain an encryption of the group member certificate along with a zero-knowledge proof that this is indeed a valid certificate issued by the group manager. Plugging a SEP in this framework is therefore straightforward and has led to very efficient constructions, e.g., [DP06,PS15]. The industrial variants of group signatures, such as DAA and EPID inherit this compatibility with SEP as they all share with group signatures the need to prove knowledge of a valid signature. It is thus no surprise that such mechanisms use SEP in practice, e.g., [BDGT17,ST21].

For a long time, no post-quantum SEP was known, except the one from [LLM$^+$16] which was mostly a proof-of-concept because of its very high complexity (see [JRS23]). Fortunately, this started to change with a very recent line of works [JRS23,BLNS23,LLLW23]. In [JRS23] the authors proposed a SEP scheme based on standard lattice assumptions leading to relatively short ZK proof of knowledge of a signature (a key building-block for most privacy-preserving mechanisms). When plugged in an anonymous credential framework, this results in a presentation transcript of about 700 KB which is a considerable improvement over [LLM$^+$16]. Soon after, [BLNS23] managed to reduce the size of this transcript to slightly under 100 KB but at the cost of relying on new ad-hoc computational assumptions. Similarly, [LLLW23] considers different security models to achieve different sizes. These approaches are then complementary as they share the same goal but with a different tradeoff between security and efficiency. The results are summarized in Table 1.1 and discussed more thoroughly in Section 8.1.

Regarding efficiency, we nevertheless note that these constructions have so far only considered the size metric which is not sufficient when we consider real-world deployments. To our knowledge, there is no public implementation of these schemes, which prevents us to assess their actual computational complexity. This is particularly problematic as they rely on intricate ZK frameworks (e.g., [LNP22]) whose performance are hard to evaluate based on their sole formal descriptions. This concretely means that, despite the relatively small sizes offered by those schemes, it is still impossible to affirm that they provide a real-world

	Assumptions	Interactive Assumption	Security	Credential Size
[JRS23]	M-SIS/M-LWE	No	Adaptive	724 KB
[BLNS23]	NTRU-ISIS$_f$	No	Adaptive	243 KB
	Int-NTRU-ISIS$_f$	Yes	Adaptive	62 KB
[LLLW23]	M-SIS/M-LWE	No	Selective	193 KB
	M-SIS/M-LWE	No	Adaptive*	372 KB
	M-SIS/M-LWE	No	Adaptive	25365 KB
Ours	M-SIS/M-LWE	No	Adaptive	80 KB

Table 1.1. Comparison of exisiting post-quantum anonymous credentials reaching 128 bits of security.

* The adaptive security proof incurs an exponential loss.

solution for the post-quantum transition of privacy-preserving authentication mechanisms.

For completeness, we also mention [BCR$^+$23] that appeared at ARES 2023. While it does provide implementation benchmarks, it is rather different from usual anonymous credential systems in many aspects. In particular, in [BCR$^+$23], each attribute is individually signed whereas usual constructions, such as those mentioned above, generate one signature on all attributes. One of the consequences is that one must now prove knowledge of one signature per attribute, which can quickly become cumbersome. Moreover, even for one attribute, the presentation transcript in [BCR$^+$23] is about 1.9 MB large, which is already much larger than those in [JRS23,BLNS23] that yet support several attributes. We will therefore not discuss [BCR$^+$23] further in this paper.

Our Contributions

In this work, we propose a construction of SEP and apply it to anonymous credentials. By revisiting the approach of [JRS23], we manage to drastically improve its performance with credentials under 80 KB, without compromising on security. We also implement our solution and show concrete practicality of our post-quantum anonymous credentials. Our natural starting point is the very recent construction of [JRS23] as it is general enough to cover most privacy-preserving use-cases while relying on standard computational assumptions. In [JRS23], a signature on a message $\mathbf{m}$ is a preimage of some syndrome $\mathbf{u}+\mathbf{Ar}+\mathbf{Dm} \bmod qR$ for a matrix $\mathbf{A_T} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]$, where:

- $\mathbf{r}$ is a short random vector chosen by the signer with a potential contribution by the user in some use-cases,
- $\mathbf{R}$ is a short trapdoor matrix,
- $\mathbf{u}$, $\mathbf{A}$, $\mathbf{AR}$ and $\mathbf{D}$ are parts of the public key,
- $\mathbf{T}$ is an invertible tag matrix,
- $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\dots|b^{k-1}] \in \mathbb{Z}^{d\times dk}$ is the gadget matrix in base $b \geq 2$, with $k = \lceil \log_b q \rceil$.

We provide the following contributions, which significantly decrease the size of the signatures and associated proofs, as illustrated in Table 8.4.

Solving the double trapdoors problem. One of the main source of inefficiency in [JRS23] is the use of statistical security arguments that requires to increase the number of columns of $\mathbf{A}$ to roughly dk and in turn the size of the signatures and of the associated zero-knowledge proofs. Our first improvement is thus to use computational security arguments based on well-studied assumptions so as to move to more compact elements and in particular smaller matrices $\mathbf{A}$ with only $2d$ columns. Far from being a mere switching of parameters, this move introduces a very technical issue that was already identified in [dPLS18,LNPS21,BLNS23] but for which no fully satisfactory solution has been proposed so far.

Let us first recall this issue. The core idea of security proofs of signature schemes based on MP trapdoors [MP12] is to change the public key so as to have a valid trapdoor for all tags but one, which we denote $\mathbf{T}^\star$. This is concretely done by replacing $\mathbf{AR}$ in the public key by $\mathbf{AR}+\mathbf{T}^\star\mathbf{G}$. As a result, for this new public key, we have $\mathbf{A_T} = [\mathbf{A}|(\mathbf{T}-\mathbf{T}^\star)\mathbf{G}-\mathbf{AR}]$ where the gadget vanishes for $\mathbf{T} = \mathbf{T}^\star$. In the computational setting, this change in the public key is done through a series of games where $\mathbf{AR}$ is first replaced by a random matrix $\mathbf{U}$ which is then replaced by $\mathbf{AR} + \mathbf{T}^\star\mathbf{G}$. At first sight, indistinguishability of these games seems to directly follow from the LWE assumption. Unfortunately, the proof is not that easy because the reduction must still produce valid signatures in the intermediate game (the one with public key $\mathbf{U}$) whereas there is no longer any trapdoor. In [dPLS18,LNPS21], this problem was solved by artificially extending the public key so as to introduce a *second* trapdoor. In the case of MP trapdoors, this concretely means using matrices of the form $\mathbf{A_T} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathbf{G} - \mathbf{AR}'] \in R_q^{2d+2kd}$ where $\mathbf{R}'$ is a second trapdoor whose only purpose is to sample preimages in this intermediate game[11]. In other words, one must almost double the dimension of the signatures because of a peculiarity of the security proof, which is quite frustrating. In [dPLS18] the authors already question the actual need for this second trapdoor whereas the ones of [BLNS23] see it as an "artifact" of the proof and propose to remove it in one of their instantiations. At this stage, we therefore end up with two unsatisfactory solutions. Either we use this redundant trapdoor to prove security or we remove it to get a more efficient scheme without security proofs.

In this paper, we show a more satisfactory solution with no compromise on security and with only a very moderate efficiency loss. We indeed leverage the specificities of preimage sampling with MP trapdoors to move from $\mathbf{AR}$ to $\mathbf{AR} + \mathbf{T}^\star\mathbf{G}$ by only replacing k columns simultaneously per game hop. More specifically, we ensure that, in each game, at most k columns of the public key have been replaced by random vectors. We therefore have, at all time, a partial trapdoor allowing to inverse all components of a syndrome but one. We then only need a way to deal with the missing component, which can be done by only

[11] In the real-world, $\mathbf{R}'$ can be discarded after having generated the public key or, alternatively, one can replace $\mathbf{G} - \mathbf{AR}'$ by a random matrix.

adding a $d \times k$ matrix $\mathbf{A}_3$ to $\mathbf{A}_\mathbf{T}$ instead of a $d \times dk$ matrix $\mathbf{AR}'$ as in the double trapdoors approach. We provide more details on this proof strategy in Section 5. As this new strategy directly decreases the dimension of the signatures, it leads to a significant improvement of their size for most[12] of the parameters we use in practice. We believe it is of independent interest, although it is very specific to MP trapdoors.

Finer Security Analysis. In the same vein as the previous improvement, we also adopt a finer analysis of the security arguments which remains statistical arguments. More precisely, we need the outputs of the Gaussian samplers to be close enough to their ideal Gaussian distributions. So far, the authors of [JRS23] only considered the statistical distance for such arguments. Other approaches based on the the Rényi divergence (say of order 2λ as suggested in [Pre17]) yield tighter security proofs and in turn more compact parameters. We thus depart from the statistical distance whenever possible. Also, as we are interested in implementing our scheme, such analyses have also proven to be beneficial to reduce the floating-point precision needed. We carry a precision analysis of our samplers and show that a precision of 53 bits is sufficient and leads to no noticeable security loss.

Removing signer's randomness. Next, we also leverage different security arguments based on rejection sampling, which is inspired from the proof technique of [CKLL19, Lem. 3.1]. The idea is to decrease the reduction loss entailed by the probability preservation property of the Rényi divergence in [JRS23], and use rejection sampling instead to only suffer a (small) constant reduction loss factor. The difference with [CKLL19] is however that we tolerate a small amount of leakage on the rejection sampling step which allows to benefit from the best of both approaches.

This modified security argument also allows for removing the randomness $\mathbf{r}$ added to the syndrome by the signer. In [JRS23], this was necessary to prove security in the chosen message setting. Although $\mathbf{r}$ can be merged with the first part of the signature, it negatively impacts the parameters as it increases the norm of this first part. Our new security reduction shows that this additional randomness is no longer necessary, which means that we can remove it altogether.

Elliptic Sampler. Another improvement comes from leveraging the elliptic sampler of [JHT22] that we revisit in Algorithm 3.2 to further reduce the signature size. This sampler stems from the observation that the perturbation used in the MP sampler is unnecessarily high for the second component of the preimage. Splitting the perturbation vector into two components with different bounds yields smaller preimages, thus resulting in smaller signatures but also smaller M-SIS bounds allowing for increased security or better parameters. More details are provided in Section 3.

Hermite Normal form. Then, using Hermite Normal Form assumptions and matrices, we can use similar tricks as [PFH+20,EFG+22,ETWY22] to reduce the

[12] More specifically, this strategy is more efficient than the one based on double trapdoors in the module case, i.e., whenever $d > 1$.

signature size without affecting security by sending only part of the signature and recovering the remaining part during verification. Unfortunately, it has no impact on the zero-knowledge proof size because one needs to recompute the full preimage to perform the proof.

Tighter bounds. Finally, we use parameter optimizations by using tighter probabilistic bounds in several places. The first stems from a better use of the Gaussian tail bound of [Ban93] to get a probability of $2^{-\lambda}$ instead of 2^{-2n} in [JRS23], where n is the dimension of the Gaussian which is usually much bigger than λ. Then, we change the distribution of the secret key from uniform $U(S_1)$ to centered binomial $\mathcal{B}_1$ as it leads to smaller spectral norms (which defines the quality of our sampler). We can also use spectral norm bounds that are satisfied only with constant probability instead of overwhelming, as long as the bound is enforced during key generation. It means that key generation might sample several secret keys until it finds a good one, and it only reduces the size of the secret key space by a constant factor. Then, at many occasions we need to bound the norm $\|\mathbf{Sx}\|_2$ for a ternary matrix $\mathbf{S}$ and a short integer vector $\mathbf{x}$. Although one could use the spectral norm of $\mathbf{S}$, it turns out to overshoot the bound we expect in practice. Instead, we use Johnson-Lindenstrauss-like bounds, as is done for example in [GHL22]. We obtain bounds of $O(\sqrt{N})\|\mathbf{x}\|_2$ instead of $O(\sqrt{N}+\sqrt{M})\|\mathbf{x}\|_2$, where N is the number of rows of $\mathbf{S}$ and M the dimension of $\mathbf{x}$. Since N is usually much smaller than M, we get much tighter bounds leading to an improved parameter selection.

Performance and Implementation. Together, these modifications yield significant improvements over [JRS23], leading to a credential under 80 KB. We note that this is the first scheme to achieve such sizes in the post-quantum setting while relying on standard and non-interactive assumptions. A comparison with existing post-quantum anonymous credentials [JRS23,BLNS23,LLLW23] is given in Table 1.1 and the full discussion is deferred to Section 8.

Finally, we implemented our anonymous credential scheme in C to evaluate its concrete performance when run on a laptop. Although our code is designed to be portable (it uses a generic arithmetic backend and does not use parallelisation), we get timings that we deem reasonable for most use-cases on this type of hardware. In particular, issuance and showing (including verification) of a credential take respectively 400 ms and 500 ms on average, values that seemed beyond reach a few years ago.

More generally, our code, which is publicly available[13], allows to better understand the actual performance of the ZK proof system from [LNP22], a powerful tool that was so far mostly used as an abstract building block. It is therefore likely to have applications outside the sole anonymous credential area, by providing a way to assess the performance of related privacy-preserving primitives such as group signatures and blind signatures.

[13] https://github.com/Chair-for-Security-Engineering/lattice-anonymous-credentials

Organization

We start by recalling the necessary notions and results in Section 2. Section 3 introduces the elliptic sampler used in our construction, and we present the trapdoor switching method in Section 4. Then, Section 5 and 6 are dedicated to presenting the signature with efficient protocols and anonymous credentials respectively. We describe the zero-knowledge arguments used in our anonymous credentials in Section 7. Finally, we present our implementation in Section 8 and carry the precision analysis of the Gaussian samplers in Section 9 necessary for our implementation.

2 Preliminaries

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

2.1 Algebraic Number Theory

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

Subring Embedding. The ring R can naturally be embedded into $\mathbb{Z}^n$, but one can generalize the embedding to the subrings of R. Using subrings can lead to interesting performance improvements in our system as our signature scheme of Section 5 is designed to interact with zero-knowledge arguments. As explained in [LNPS21], using a ring of smaller degree allows for reducing the proof size. This is however at the expense of a lower compression of the keys for the signature scheme. A solution to obtain the best of both worlds is to use a ring R of degree n for the signature, and a subring $\widehat{R}$ of degree $\widehat{n}|n$ for the zero-knowledge proof. This requires embedding the relations over R into relations over $\widehat{R}$ (in turn increasing the dimension by $\widehat{k} = n/\widehat{n}$). Although this is folklore algebra and already used implicitly in [LNPS21,LNP22], we give for completeness all the algebraic details needed to map R to $\widehat{R}$. We let $\widehat{n}|n$ be a power of two, and $\widehat{k} = n/\widehat{n}$. In addition to R, we define $\widehat{R} = \mathbb{Z}[x]/\langle x^{\widehat{n}} + 1\rangle$. To avoid confusion, when relevant and not clear from the context, we use $\otimes_R$ to denote the product in R, and $\otimes_{\widehat{R}}$ for the product in $\widehat{R}$.

Even though there are many ways to embed R into $\widehat{R}^{\widehat{k}}$, we define the embedding $\theta : R \to \widehat{R}^{\widehat{k}}$ as follows. For $a = \sum_{\ell\in[0,n-1]} a_\ell x^\ell \in R$ with $(a_\ell)_\ell \in \mathbb{Z}^n$, and for all $i \in [0, \widehat{k}-1]$, define $\widehat{a}_i = \sum_{j\in[0,\widehat{n}-1]} a_{\widehat{k}j+i} x^j \in \widehat{R}$. Then, the embedding of a is defined by $\theta(a) = [\widehat{a}_0|\dots|\widehat{a}_{\widehat{k}-1}]^T \in \widehat{R}^{\widehat{k}}$.

This embedding relies on the fact that a can be uniquely written as $a = \sum_{i\in[0,\widehat{k}-1]} \sum_{j\in[0,\widehat{n}-1]} a_{\widehat{k}j+i} x^{\widehat{k}j+i}$, which itself equals $\sum_{i\in[0,\widehat{k}-1]} \widehat{a}_i(x^{\widehat{k}}) \otimes_R x^i$. This in particular defines the inverse embedding θ^{-1}.

Operations and Multiplication Matrix. The embedding θ (and its inverse) is clearly linear, which means that addition in R can be performed over $\widehat{R}^{\widehat{k}}$ coefficient-wise and vice-versa. In [LNPS21, Lem. 2.11], Lyubashevsky et al. recall that the multiplication $a \otimes_R b$ can also be performed on the embeddings $\theta(a), \theta(b)$ using a carefully defined multiplication $\otimes_{\widehat{R}^{\widehat{k}}} : \widehat{R}^{\widehat{k}} \times \widehat{R}^{\widehat{k}} \to \widehat{R}^{\widehat{k}}$, that can be carried using only additions and $\otimes_{\widehat{R}}$. For two elements $a, b \in R$ such that $\theta(a) = [\widehat{a}_0|\dots|\widehat{a}_{\widehat{k}-1}]^T$ and $\theta(b) = [\widehat{b}_0|\dots|\widehat{b}_{\widehat{k}-1}]^T$, we have $\theta(a) \otimes_{\widehat{R}^{\widehat{k}}} \theta(b) = [\widehat{c}_0|\dots|\widehat{c}_{\widehat{k}-1}]^T$, where

$$\widehat{c}_\ell = \sum_{\substack{0\le i,j<\widehat{k} \\ i+j=\ell \bmod \widehat{k}}} \widehat{a}_i \otimes_{\widehat{R}} \widehat{b}_j \otimes_{\widehat{R}} x^{\left\lfloor \frac{i+j}{\widehat{k}} \right\rfloor},$$

for all $\ell \in [0, \widehat{k}-1]$. We can simplify this expression by observing that for a fixed $j \in [0, \widehat{k}-1]$, there is only one $i \in [0, \widehat{k}-1]$ verifying $i + j = \ell \bmod \widehat{k}$, namely $i = \ell - j$ if $\ell \ge j$, and $i = \ell - j + \widehat{k}$ otherwise. We thus get

$$\begin{aligned}\widehat{c}_\ell &= \sum_{j=0}^{\ell} \widehat{a}_{\ell-j} \otimes_{\widehat{R}} \widehat{b}_j + \sum_{j=\ell+1}^{\widehat{k}-1} \widehat{a}_{\ell-j+\widehat{k}} \otimes_{\widehat{R}} x \otimes_{\widehat{R}} \widehat{b}_j \\ &= [\widehat{a}_\ell|\dots|\widehat{a}_0|\widehat{a}_{\widehat{k}-1}x|\dots|\widehat{a}_{\ell+1}x] \cdot \theta(b).\end{aligned}$$

This rewriting highlights the expression of a multiplication matrix $M_\theta(a)$ so that $\theta(a \otimes_R b) = \theta(a) \otimes_{\widehat{R}^{\widehat{k}}} \theta(b) = M_\theta(a)\theta(b)$ where the latter matrix-vector product is performed in $\widehat{R}$. Formally, we have

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

Another useful way to express $M_\theta(a)$ is by observing that for $i \in [0, \widehat{k}-1]$, the i-th column of $M_\theta(a)$ corresponds to $\theta(a \otimes_R x^i)$. Hence $M_\theta(a) = [\theta(a)|\theta(a \otimes_R x)|\dots|\theta(a \otimes x^{\widehat{k}-1})]$. We naturally extend the embedding θ to vectors and the multiplication map M_θ blockwise to vectors and matrices over R, i.e., for $\mathbf{A} = [a_{i,j}]_{i,j} \in R^{d\times m}$ by $M_\theta(\mathbf{A}) = [M_\theta(a_{i,j})]_{i,j} \in \widehat{R}^{\widehat{k}d\times\widehat{k}m}$.

Coefficient Embedding. A specific case of the subring embedding θ is when the subring $\widehat{R}$ is of degree 1. In this case, we are considering $\widehat{n} = 1$, $\widehat{k} = n$ and $\widehat{R} = \mathbb{Z}[x]/\langle x + 1\rangle = \mathbb{Z}$. It is then called coefficient embedding and we denote it by τ to avoid confusion. This corresponds to mapping ring elements to their coefficient vectors, i.e., for all $a = \sum_{i\in[0,n-1]} a_i x^i \in R$, $\tau(a) = [a_0|\ldots|a_{n-1}]^T$. We can also consider the associated multiplication matrix map, which we call M_τ, that is defined as in Equation (1). The difference is that in this ring of degree 1, x is equal to -1, thus yielding

$$M_\tau(a) = \begin{bmatrix} a_0 & -a_{n-1} & \ldots & -a_1 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{n-1} \\ a_{n-1} & \ldots & a_1 & a_0 \end{bmatrix},$$

so that for all $a, b \in R$, $\tau(ab) = M_\tau(a)\tau(b) \in \mathbb{Z}^n$. We also extend τ to vectors of R^d by concatenating the coefficient embeddings of each vector entry, and M_τ blockwise to matrices over R. Then, for an integer η, we define $S_\eta = \tau^{-1}([-\eta, \eta]^n)$ and $T_\eta = \tau^{-1}([0, \eta]^n)$. We also define the usual vector norms $\|\cdot\|_p$ over R by $\|r\|_p := \|\tau(r)\|_p$, and the spectral norm $\|\mathbf{A}\|_2$ by $\|M_\tau(\mathbf{A})\|_2$.

The coefficient embedding can be defined with respect to R but also with respect to a subring $\widehat{R}$ of R. If needed, we differentiate them by τ_R and $\tau_{\widehat{R}}$. When both are present, we use $\widehat{S}_\eta$ and $\widehat{T}_\eta$ for the corresponding sets S_η and T_η but with respect to the subring $\widehat{R}$.

Remark 2.1 (Coefficient Embedding and Subrings). For an element a of R expressed as $a = \sum_{\ell\in[0,n-1]} a_\ell x^\ell$, we define only here $a_{i,j} = a_{\widehat{k}j+i}$. Then, we have

$$\begin{aligned} \tau_R(a) &= [a_{0,0}|\ldots|a_{\widehat{k}-1,0}|\ldots|a_{0,\widehat{n}-1}|\ldots|a_{\widehat{k}-1,\widehat{n}-1}]^T \\ \tau_{\widehat{R}}(\theta(a)) &= [a_{0,0}|\ldots|a_{0,\widehat{n}-1}|\ldots|a_{\widehat{k}-1,0}|\ldots|a_{\widehat{k}-1,\widehat{n}-1}]^T, \end{aligned}$$

which means that embedding through θ only permutes the coefficients. More precisely, we define the permutation matrix $\mathbf{P}_\theta$ by its entries being $[\mathbf{P}_\theta]_{\widehat{k}j+i,\widehat{n}i+j} = 1$ for all $(i, j) \in [0, \widehat{k} - 1] \times [0, \widehat{n} - 1]$, and 0 elsewhere. We thence have $\tau_R(a) = \mathbf{P}_\theta \cdot \tau_{\widehat{R}}(\theta(a))$. This implies that proving statements such as $\tau_R(a) \in \{0, 1\}^n$ (i.e. $a \in T_1$) or $\|\tau_R(a)\|_p \leq B$ is strictly equivalent to proving these statement on $\theta(a)$ over $\widehat{R}^{\widehat{k}}$, that is $\tau_{\widehat{R}}(\theta(a)) \in \{0, 1\}^n$ or $\left\|\tau_{\widehat{R}}(\theta(a))\right\|_p \leq B$.

Conjugate. We later use the conjugate a^* of a ring element $a \in R$. More precisely, we define $a^* = a(x^{-1})$ which in the case of power-of-two cyclotomic rings equals $a_0 - \sum_{i\in[n-1]} a_{n-i} x^i$. For a matrix $\mathbf{A} = [a_{i,j}]_{i,j} \in R^{d\times m}$, we define $\mathbf{A}^* = [a^*_{j,i}]_{i,j}$ which is the conjugate transpose. We use the same notations when working over a subring $\widehat{R}$ and keep the latter implicit in the notation.

2.2 Lattices

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

2.3 Probabilities

For a finite set S, we define $|S|$ to be its cardinality, and $U(S)$ to be the uniform probability distribution over S. We also let ψ_η be the centered binomial distribution of parameter $\eta \in \mathbb{N} \setminus \{0\}$ defined by the distribution of $\sum_{i\in[\eta]} a_i - b_i$ for $a_1, b_1, \ldots, a_\eta, b_\eta$ independently drawn from $U(\{0,1\})$. We then use $\mathcal{B}_\eta$ to denote the distribution over R whose coefficients follow ψ_η, that is $\tau^{-1}(\psi_\eta^n)$. We use $x \hookleftarrow \mathcal{P}$ to describe the action of sampling $x \in S$ according to the probability distribution $\mathcal{P}$. In contrast, we use $x \sim \mathcal{P}$ to mean that the random variable x follows $\mathcal{P}$. The *statistical distance* between two discrete distributions $\mathcal{P}, \mathcal{Q}$ over a countable set S is defined as $\Delta(\mathcal{P}, \mathcal{Q}) = \frac{1}{2}\sum_{x\in S}|\mathcal{P}(x) - \mathcal{Q}(x)|$. If $\mathcal{P}$ and $\mathcal{Q}$ are such that the support of $\mathcal{P}$, denoted by S, is a subset of that of $\mathcal{Q}$, then we define the Rényi divergence of order $a \in (1, \infty)$ from a $\mathcal{P}$ to $\mathcal{Q}$ is defined by $RD_a(\mathcal{P}\|\mathcal{Q}) = (\sum_{\mathbf{x}\in S} \mathcal{P}(\mathbf{x})^a/\mathcal{Q}(\mathbf{x})^{a-1})^{1/(a-1)}$. The Rényi divergence of infinite order from $\mathcal{P}$ to $\mathcal{Q}$ is $RD_\infty(\mathcal{P}\|\mathcal{Q}) = \max_{\mathbf{x}\in S} \mathcal{P}(\mathbf{x})/\mathcal{Q}(\mathbf{x})$. We give the following lemma which simply combines the probability preservation from [BLR+18] and the relative error lemma from [Pre17].

Lemma 2.1 ([BLR+18, Lem. 2.9][Pre17, Lem. 3]). *Let $\mathcal{P}_1, \mathcal{P}_2$ be two distributions having the same support. Let $\delta > 0$ be such that $\forall x \in Supp(\mathcal{P}_1)$, $1 - \delta \leq \mathcal{P}_1(x)/\mathcal{P}_2(x) \leq 1 + \delta$. Then, for all $a \in (1, \infty)$ and all events $E \subseteq Supp(\mathcal{P}_1)$, it holds that*

$$\mathcal{P}_1(E) \leq \left(1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}}\right)^{1/a} \mathcal{P}_2(E)^{(a-1)/a} \underset{\delta\to 0}{\sim} \left(1 + \frac{a-1}{2}\delta^2\right) \mathcal{P}_2(E)^{(a-1)/a}$$

Probabilistic Norm Bounds. The quality of our preimage sampler depends on the spectral norm of our secret key which we need to bound. For that we rely on the following (heuristic) bound inspired by the proven bound of [Ver12] in the case of unstructured matrices. We note that even though it does not fit the exact requirements of [Ver12], this bound has been extensively used, e.g. [MP12,GMPW20,LNP22], and verified by our experiments.

Lemma 2.2 (Heuristic). *Let d, m be two positive integers. It (heuristically) holds that $\mathbb{P}_{\mathbf{R}\sim\mathcal{B}_1^{d\times m}}[\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{d} + \sqrt{m} + 6)] = 1/O(1)$ (in particular non-negligible).*

We also use the following Johnson-Lindenstrauss-type bound stating that for an arbitrary vector $\mathbf{m}$ and a random short matrix $\mathbf{S}$, then $\mathbf{S}\mathbf{m}$ is not significantly larger than **m** except with negligible probability. We prove the first following lemma and then provide a tighter bound which is backed up by experiments. The first lemma generalizes the bound provided in [JRS23, Lem. 2.5].

Lemma 2.3. *Let d, m be two positive integers and $\lambda > 0$. Let $\mathbf{m} \in \mathbb{Z}^m$ and $\mathcal{P}$ be a distribution with subgaussian moment $s > 0$. Then it holds that*

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{P}^{d\times m}}\left[\|\mathbf{Sm}\|_2 \geq \sqrt{4+2\sqrt{\frac{\lambda}{d}}\left(\sqrt{\frac{\lambda}{d}}+\sqrt{\frac{8}{\ln 2}+\frac{\lambda}{d}}\right)\ln 2}\cdot s\sqrt{d}\|\mathbf{m}\|_2\right] \leq 2^{-\lambda}.$$

Proof. Define $\beta = \|\mathbf{m}\|_2$, and s be the subgaussian moment of $\mathcal{P}$. Let $\mathbf{S} \sim \mathcal{P}^{d\times m}$. Let $i \in [d]$ and $t \in \mathbb{R}$. Then,

$$\begin{aligned}
\mathbb{E}_{\mathbf{s}_i}[\exp(t\mathbf{s}_i^T\mathbf{m})] &= \mathbb{E}_{\mathbf{s}_i}\left[\prod_{j\in[m]} e^{tm_j s_{i,j}}\right] \\
&= \prod_{j\in[m]} \mathbb{E}_{s_{i,j}}[\exp(tm_j s_{i,j})] \\
&\leq \prod_{j\in[m]} \exp(s^2(tm_j)^2/2) \\
&= \exp((\beta s)^2 t^2/2).
\end{aligned}$$

So $x_i = \mathbf{s}_i^T\mathbf{m}_i$ is βs-subgaussian for each $i \in [d]$. Let $y_i = x_i^2$ and $\mu_i = \mathbb{E}_{\mathbf{s}_i}[y_i]$. Because x_i is βs-subgaussian, it means that

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

In particular, $\mu_i \leq 2(\sqrt{2}\beta s)^2\Gamma(1) = 4\beta^2 s^2$. As a consequence, it holds that

$$\begin{aligned}
\mathbb{E}_{\mathbf{s}_i}[\exp(t(y_i-\mu_i))] &= 1 + t\mathbb{E}_{\mathbf{s}_i}[y_i-\mu_i] + \sum_{p\geq 2}\frac{t^p}{p!}\mathbb{E}_{\mathbf{s}_i}[(x_i^2-\mu_i)^p] \\
&\leq 1 + \sum_{p\geq 2}\frac{t^p}{p!}\mathbb{E}_{\mathbf{s}_i}[x_i^{2p}] \\
&\leq 1 + \sum_{p\geq 2}\frac{t^p}{p!}(2p(\sqrt{2}\beta s)^{2p}\Gamma(p)) \\
&= 1 + 2\sum_{p\geq 2}(2t\beta^2 s^2)^p \\
&= 1 + 2\left(\frac{1}{1-2\beta^2 s^2 t} - (1+2\beta^2 s^2 t)\right) \\
&= 1 + \frac{8t^2\beta^4 s^4}{1-2\beta^2 s^2 t}.
\end{aligned}$$

where the second to last equality holds if $|t| \leq 1/(2\beta^2 s^2)$. So for some $\alpha \geq 1$, and for all t such that $|t| < 1/(2\alpha\beta^2 s^2)$, we have

$$\mathbb{E}_{\mathbf{s}_i}[\exp(t(y_i - \mu_i))] \leq \exp\left(\frac{16\beta^4 s^4 \alpha}{\alpha - 1} \cdot \frac{t^2}{2}\right),$$

meaning that $y_i - \mu_i$ is a centered sub-exponential random variable with parameters $\gamma = 4\beta^2 s^2 \sqrt{\alpha/(\alpha-1)}$ and $\delta = 2\beta^2 s^2 \alpha$. Thence, $y - \mu = \sum_{i \in [d]} y_i - \mu_i$ is subexponential with parameters $\gamma' = \gamma\sqrt{d}$ and $\delta' = \delta$. Using the sub-exponential tail bound, we obtain that for all $r \in (0, \gamma'^2/\delta')$,

$$\mathbb{P}_{\mathbf{S}}[y - \mu \geq r] \leq \exp(-r^2/(2\gamma'^2)).$$

This can be re-written as follows. For all $\lambda \in (0, \frac{2d}{\alpha(\alpha-1)\ln 2})$, it holds that

$$\mathbb{P}_{\mathbf{S}}\left[\|\mathbf{Sm}\|_2^2 \geq 4d\beta^2 s^2 \left(1 + \sqrt{\frac{2\alpha \ln 2}{\alpha - 1} \cdot \frac{\lambda}{d}}\right)\right] \leq 2^{-\lambda}.$$

We now fix λ and d and optimize over α. More precisely, need to maximize $\alpha > 1$ while ensuring that $\lambda < 2d/(\alpha(\alpha-1)\ln 2)$. The optimal value is then

$$\alpha^* = \frac{1}{2}\left(1 + \sqrt{1 + \frac{8d}{\lambda \ln 2}}\right),$$

We then obtain a bound on $\|\mathbf{Sm}\|_2^2/(d\beta^2 s^2)$ as

$$\gamma = 4\left(1 + \sqrt{\frac{2\alpha^* \ln 2}{\alpha^* - 1} \cdot \frac{\lambda}{d}}\right) = 4 + 2\ln 2 \cdot \sqrt{\frac{\lambda}{d}} \cdot \left(\sqrt{\frac{\lambda}{d}} + \sqrt{\frac{8}{\ln 2} + \frac{\lambda}{d}}\right).$$

We then conclude that $\mathbb{P}_{\mathbf{S} \sim \mathcal{P}^{d \times m}}[\|\mathbf{Sm}\|_2 \geq \sqrt{\gamma} \cdot s\sqrt{d}\|\mathbf{m}\|_2] \leq 2^{-\lambda}$ as desired. In general, d is much larger than λ, meaning that the factor in front of $s\sqrt{d}\|\mathbf{m}\|_2$ can be bounded by a constant, and goes to 2 for smaller ratios λ/d. □

The bound on $\|\mathbf{Sm}\|_2$ from Lemma 2.3 is only needed in the proof of unforgeability of our signature. As a result, it only needs to be verified with a probability that is non-negligible, say a constant, but it does not have to be overwhelming[14]. For example, if the bound is verified only with a probability of 1/2, it only entails a couple of extra bits in the security loss. This allows us to obtain tighter bounds and in turn tighter parameter constraints. We note that such results are obtained with overwhelming probability in [GHL22,LNP22] based on the normal-distribution heuristic but the latter is not verified for structured matrices. This is why we provide the following bound which is empirically verified in the structured case.

[14] Similar bounds for unstructured matrices are used in the zero-knowledge proof system. The (heuristical) bound of [LNP22, Lem. 2.8] is $\sqrt{337}\|\mathbf{m}\|_2$ for $\mathcal{P} = \psi_1$ and $(d, \lambda) = (256, 128)$. In this case, we need an overwhelming probability. For the same parameters, our proven result yields a bound of $\sqrt{1037}\|\mathbf{m}\|_2$ instead.

Lemma 2.4 (Heuristic). *Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power-of-two. Let d, m be two positive integers. For an arbitrary $\mathbf{m} \in R^m$, it heuristically holds that $\mathbb{P}_{\mathbf{S}\sim\mathcal{B}_1^{d\times m}}[\|\mathbf{Sm}\|_2 \leq \frac{1}{\sqrt{2}}\sqrt{nd}\|\mathbf{m}\|_2] = 1/C$ with $C = O(1)$ (in particular non-negligible).*

More generally, we observe that $\mathbb{P}_{\mathbf{S}\sim\mathcal{B}_1^{d\times m}}[\|\mathbf{Sm}\|_2 \leq \gamma\sqrt{nd}\|\mathbf{m}\|_2]$ is negligible for some γ that is at most 1 for typical values of n and d, and such that $\lim_{nd\to\infty} \gamma = 1/\sqrt{2}$. It is also empirically verified for other centered subgaussian distributions but where $\lim_{nd\to\infty} \gamma = s$, s denoting the subgaussian moment of the distribution.

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

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

We will need a Gaussian regularity lemma from [GPV08, Lem. 5.2] generalized to non-spherical distributions. We state it over rings for coherence but it also applies to the integers. We first prove the generalized version of [GPV08, Cor. 2.8] stated here.

Lemma 2.5 ([GPV08, Cor. 2.8] adapted). *Let d be a positive integer, and $\mathcal{L}' \subseteq \mathcal{L} \subset \mathbb{R}^d$ be two full rank lattices. Then, let $\varepsilon \in (0, 1)$, $\mathbf{S} \in \mathbb{R}^{d\times d}$ be such that $\mathbf{S} - \eta_\varepsilon(\mathcal{L}')^2\mathbf{I}_d$ is positive semi-definite, and $\mathbf{c} \in \mathbb{R}^d$. We denote by $\mathcal{P}_0 = \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}} \bmod \mathcal{L}'$ and $\mathcal{P}_1 = U(\mathcal{L} \bmod \mathcal{L}')$. It then holds that*

$$\forall \mathbf{x} \in \mathcal{L} \bmod \mathcal{L}', \mathcal{P}_0(\mathbf{x}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right].$$

When $\mathbf{c} = \mathbf{0}$, we have $\mathcal{P}_0(\mathbf{x}) \in [(1-\varepsilon)/(1+\varepsilon), 1+\varepsilon]\mathcal{P}_1(\mathbf{x})$.

Proof. Let $\mathbf{z}$ be distributed according to $\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}$. Let $\mathbf{v} + \mathcal{L}'$ be a coset of $\mathcal{L}/\mathcal{L}'$. Then, it holds that

$$\mathbb{P}_\mathbf{z}[\mathbf{z} = \mathbf{v} \bmod \mathcal{L}'] = \frac{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{v} + \mathcal{L}')}{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L})}.$$

By Poisson's summation formula and our condition on $\mathbf{S}$, it holds that $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{v} + \mathcal{L}') = \rho_{\sqrt{\mathbf{S}},\mathbf{c}-\mathbf{v}}(\mathcal{L}') \in \sqrt{\det \mathbf{S}}(\text{Vol } \mathcal{L}')^{-1}[1-\varepsilon, 1+\varepsilon]$. Similarly, because $\eta_\varepsilon(\mathcal{L}') \geq$

$\eta_\varepsilon(\mathcal{L})$, we get $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L}) \in \sqrt{\det \mathbf{S}}(\mathrm{Vol}\ \mathcal{L})^{-1}[1-\varepsilon, 1+\varepsilon]$ (it becomes $[1, 1+\varepsilon]$ when $\mathbf{c} = \mathbf{0}$). As a result, we have

$$\mathbb{P}_{\mathbf{z}}[\mathbf{z} = \mathbf{v} \bmod \mathcal{L}'] \in \frac{\mathrm{Vol}\ \mathcal{L}}{\mathrm{Vol}\ \mathcal{L}'}\left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] = \frac{1}{|\mathcal{L}/\mathcal{L}'|}\left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right],$$

as desired. □

Lemma 2.6 ([GPV08, Lem. 5.2] adapted). *Let d, m, q be positive integers, and $\overline{\mathbf{A}} \in R_q^{d \times m}$ such that $\overline{\mathbf{A}} R_q^m = R_q^d$. Then, let $\varepsilon \in (0,1)$ and $\mathbf{S} \in \mathbb{R}^{nm \times nm}$ such that $\mathbf{S} - \eta_\varepsilon(\mathcal{L}_q^\perp(\overline{\mathbf{A}}))^2 \mathbf{I}_{nm}$ is positive semi-definite. We finally define $\mathcal{P} = \overline{\mathbf{A}}\mathcal{D}_{R^m,\sqrt{\mathbf{S}}} \bmod qR$. It holds that $\forall \mathbf{x} \in R_q^d, \mathcal{P}(\mathbf{x}) \in [(1-\varepsilon)/(1+\varepsilon), 1+\varepsilon]q^{-nd}$.*

Proof. It clearly holds that its support is $\overline{\mathbf{A}}R^m \bmod qR = \overline{\mathbf{A}}R_q^m = R_q^d$. Applying Lemma 2.5 to $\mathcal{L} = R^m$ and $\mathcal{L}' = \mathcal{L}_q^\perp(\overline{\mathbf{A}})$ (through their embedding to $\mathbb{Z}^{nm}$), we directly obtain that $\forall \mathbf{x} \in R_q^d, \mathcal{P}(\mathbf{x}) \in [(1-\varepsilon)/(1+\varepsilon), 1+\varepsilon] \cdot |\mathcal{L}/\mathcal{L}'|^{-1}$. Yet $\mathcal{L}/\mathcal{L}'$ is isomorphic to $\overline{\mathbf{A}}R^m \bmod qR = R_q^d$ so $|\mathcal{L}/\mathcal{L}'| = q^{nd}$ as desired. □

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

Lemma 2.7 ([Ban93, Lem. 1.5]). *Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d, and $s > 0$. Then, for all $c > 1/\sqrt{2\pi}$, it holds that*

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

Typically, for $c = 1$, the probability is at most 2^{-2d} which is a little too conservative. Instead, we later use a slack c to tweak the tailcut probability. To be more precise, c now denotes a function that takes the dimension d as input and a parameter λ, and outputs the smallest $c > 1/\sqrt{2\pi}$ such that $(c\sqrt{2\pi e}e^{-\pi c^2})^d \leq 2^{-\lambda}$. For example, it holds for any dimension d that $c(d,d) \approx 0.767$. As an other example, we have $c(512, 128) \approx 0.5751$. For clarity, we simply use c_d to denote $c(d, \lambda + O(1))$ where λ is the security parameter. Heuristically, we could even choose $c_d = 1/\sqrt{2\pi} \approx 0.4$ and have the bound verified with high probability.

Rejection Sampling. We first give the rejection sampling results from [DFPS22, Lem. 2.2, Lem. 4.1, Lem. C.2] which are needed in the zero-knowledge arguments. It makes use of the algorithm Rej_1 from Algorithm 2.1.

Algorithm 2.1: $\mathsf{Rej}_1(\mathbf{z}, \mathbf{s}, s, M)$

```
1. $u \hookleftarrow U([0,1))$.
2. return 1 if $u \leq \frac{1}{M}\exp\left(\frac{\pi}{s^2}(\|\tau(\mathbf{s})\|_2^2 - 2\langle\tau(\mathbf{z}), \tau(\mathbf{s})\rangle)\right)$, and 0 otherwise.
```

Lemma 2.8 (Adapted from [DFPS22, Lem. 2.2, Lem. 4.1, Lem. C.2]). *Let d be a positive integer. Let $S \subset R^d$ be a set of vectors of ℓ_2 norm at most $T > 0$, and $\mathscr{D}_S$ be a distribution over S. Let $M > 1$, $\varepsilon \in (0, 1/2]$ and let $\alpha = \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}})$. Then, let $s \geq \alpha T$. We define the following distributions.*

$\mathcal{P}_1$ *Sample* $\mathbf{s} \hookleftarrow \mathscr{D}_S$, $\mathbf{y} \hookleftarrow \mathcal{D}_{R^d,s}$ *and set* $\mathbf{z} = \mathbf{y} + \mathbf{s}$. *Then let* $b \leftarrow \mathsf{Rej}_1(\mathbf{z}, \mathbf{s}, s, M)$. *If* $b = 1$, *output* $(\mathbf{s}, \mathbf{z})$, *and* $\perp$ *otherwise.*

$\mathcal{P}_2$ *Sample* $\mathbf{s} \hookleftarrow \mathscr{D}_S$ *and* $\mathbf{z} \hookleftarrow \mathcal{D}_{R^d,s}$. *Then sample a continuous* $u \hookleftarrow U([0,1))$. *If* $u \leq 1/M$, *output* $(\mathbf{s}, \mathbf{z})$, *and* $\perp$ *otherwise.*

Then, $\varDelta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon/M$, *and* $RD_\infty(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1 + \varepsilon/(M-1)$.

We also need another rejection sampling result from [LNS21] which leaks at most one bit of information if it is to hide ephemeral randomness. It is similar to the previous one except that it also rejects based on the direction of $\mathbf{z}$ with respect to $\mathbf{s}$. Note it cannot be used for long-term secrets as leakage would increase with repetition.

Lemma 2.9 ([LNS21, Lem. 3.2]). *Let d be a positive integer. Let $S \subset R^d$ be a set of vectors of ℓ_2 norm at most $T > 0$, and $\mathscr{D}_S$ be a distribution over S. Let $M > 1$ and $\alpha = \sqrt{\pi/\ln M}$. Then, let $s \geq \alpha T$. We define the following distributions.*

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

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

Then, $\mathcal{P}_1$ outputs $(\mathbf{s}, \mathbf{z}) \neq \perp$ with probability at least $1/2M$, and conditioned on not aborting it holds that $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical.

2.4 Hardness Assumptions

The security of the signature of Section 5 is based on the *Module Short Integer Solution* (M-SIS) and *Module Learning With Errors* (M-LWE) problems [LS15], which we now recall. We consider both problems in their Hermite Normal Form, i.e., we specify the identity in the M-SIS matrix, and we use the same distribution for the M-LWE secret and error.

Definition 2.1 (M-SIS). *Let $R = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ with n a power-of-two. Let d, m, q be positive integers and $\beta > 0$ with $m > d$. The Module Short Integer Solution problem in Hermite Normal Form* $\text{M-SIS}_{n,d,m,q,\beta}$ *asks to find $\mathbf{x} \in \mathcal{L}_q^\perp([\mathbf{I}_d | \mathbf{A}']) \setminus \{\mathbf{0}\}$ such that $\|\mathbf{x}\|_2 \leq \beta$, given $\mathbf{A}' \hookleftarrow U(R_q^{d \times m - d})$.*

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

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

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

Definition 2.2 (M-LWE). *Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ with n a power-of-two. Let d, m, k, q be positive integers and $\mathscr{D}_r$ a distribution on R. The Module Learning With Errors problem* $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$ *asks to distinguish between the following distributions: (1)* $(\mathbf{A}', [\mathbf{I}_m|\mathbf{A}']\mathbf{R} \bmod qR)$*, where* $\mathbf{A}' \sim U(R_q^{m\times d})$ *and* $\mathbf{R} \sim \mathscr{D}_r^{d+m\times k}$*, and (2)* $(\mathbf{A}', \mathbf{B})$*, where* $\mathbf{A}' \sim U(R_q^{m\times d})$ *and* $\mathbf{B} \sim U(R_q^{m\times k})$*.*

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

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

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

2.5 Digital Signature

A digital signature corresponds to a collection of four algorithms Setup, KeyGen, Sign, Verify. The Setup algorithm generates the public parameters pp from the security parameter λ. Then, each signer runs the KeyGen algorithm and obtain a secret key sk for signing, and a public key pk for verification. The Sign algorithm takes a message $\mathbf{m}$ and a signing key sk (and sometimes pp, pk) to produce a signature sig. Finally, from the message $\mathbf{m}$, the signature sig and the verification key pk (and pp), Verify outputs 1 if sig is a valid signature on $\mathbf{m}$ for the key pk, and 0 otherwise. The signer can also maintain a state st which is used to keep track of some information necessary for the signing procedure. The state can be as simple as a counter, but can also be more complex like a table storing all the previously emitted signatures.

Security-wise, we expect the digital signature to be correct, that is that honestly generated signatures produced by honest keys pass verification with high probability. Formally, we expect that for any $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, and any message $\mathbf{m}$, $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, \mathsf{Sign}(\mathsf{sk}, \mathbf{m}, \mathsf{pk}, \mathsf{pp}), \mathsf{pp}) = 1$. We also expect unforgeability which captures the fact that an adversary able to see or request signatures on message of their choice cannot produce a valid signature on a new message without knowing the signing key. It is modeled by the security game from Figure 2.1. If the probability that the polynomial adversary wins the corresponding security game is negligible in λ, the signature is said to be *existentially unforgeable against chosen-message attacks* or EUF-CMA secure.

2.6 Anonymous Credentials

An anonymous credential system can essentially be seen as an elaborate digital signature scheme where an *organization* generates credentials on attributes for *users* through an interactive process $\mathsf{Issue}_{O,U}$. The credentials can thus be shown to *verifiers* through an interactive protocol $\mathsf{Show}_{U,V}$. In other words, $\mathsf{Issue}_{O,U}$ and $\mathsf{Show}_{U,V}$ can be seen as the interactive counterparts of Sign and

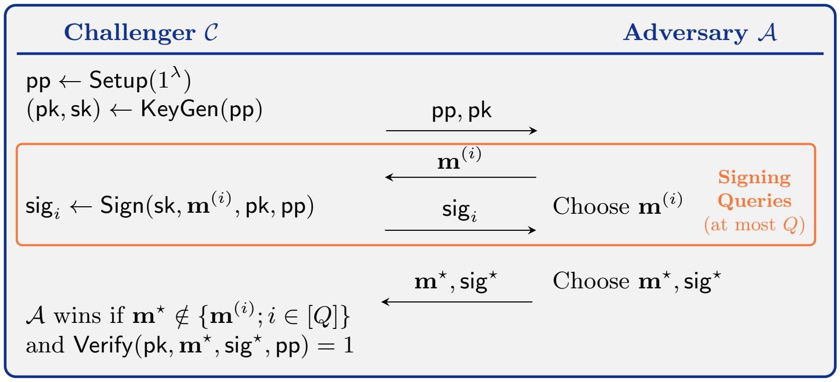
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
pp, pk
$\mathbf{m}^{(i)}$
$\mathsf{sig}_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{m}^{(i)}, \mathsf{pk}, \mathsf{pp})$
sig_i
Choose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

Fig. 2.1. Existential Unforgeability against Chosen Message Attacks game

Verify algorithms which handle *credentials* and *attributes* instead of *signatures* and *messages*. Anonymous credentials extend the EUF-CMA security notion to this context but primarily introduce privacy considerations that do not exist for standard digital signatures.

As in [JRS23], we use the formal definition and security model from [FHS19] that we formally recall below. This model defines two main security properties, namely unforgeability and anonymity. The former essentially requests that no adversary can convince a verifier that it owns a valid credential on a set of attributes if it has not received such a credential from the organization. Anonymity requires that no one, even the organization, can identify the user running the $\mathsf{Show}_{U,V}$ protocol unless the set of disclosed attributes trivially allows it to do so. This means that no information leaks on the credential nor on the undisclosed attributes.

We now formally describe the necessary notations and security definitions. An anonymous credentials system is a collection of four algorithms $\mathsf{OKeyGen}$, $\mathsf{UKeyGen}$, $\mathsf{Issue}_{O,U}$, $\mathsf{Show}_{U,V}$, where $\mathsf{OKeyGen}$ and $\mathsf{UKeyGen}$ take public parameters and output the organization's and user's key pairs respectively, which are denoted by $(\mathsf{opk}, \mathsf{osk})$ and $(\mathsf{upk}, \mathsf{usk})$. Then, $\mathsf{Issue}_{O,U}$ is an interactive protocol between the organization O holding $(\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st})$ and the users' attributes $\mathbf{m}$, and a user U holding $(\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp})$ and its attributes $\mathbf{m}$. The user receives a credential cred on $\mathbf{m}$ if the protocol went through, while O only knows whether or not the execution was successful. Finally, $\mathsf{Show}_{U,V}$ is a (possibly non-interactive) protocol between a user U with $(\mathsf{usk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}, \mathsf{cred}, \mathcal{I})$ and a verifier V having $(\mathsf{opk}, \mathsf{pp}, \mathbf{m}_\mathcal{I})$. It outputs 1 to V if cred is valid for the disclosed attributes $\mathbf{m}_\mathcal{I}$ and 0 otherwise, while U gets no output.

We require such a system to be correct, anonymous and unforgeable. The latter two properties are defined using the following notations.

- HU: Set of user indices of honest users ($\varnothing$ at the outset).
- CU: Set of user indices of corrupt users ($\varnothing$ at the outset).
- ctr: Issuance counter (0 at the outset).

- A: Set of triplets $(j, j', (\mathbf{m}_i)_{i\in[\ell]})$ filled after a successful issuance of credentials for user j on attributes $\mathbf{m}$ and issuance index j' ($\mathcal{O}_{\mathsf{ObtIss}}$ or $\mathcal{O}_{\mathsf{Issue}}$).
- $\mathcal{O}_{\mathsf{HU}}(j)$: Given a user index j, it returns $\perp$ if $j \in \mathsf{HU} \cup \mathsf{CU}$. Otherwise, it samples $(\mathsf{upk}_j, \mathsf{usk}_j) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$, adds j to HU and returns upk_j.
- $\mathcal{O}_{\mathsf{CU}}(j, \mathsf{upk})$: Given a user index j and optionally a public key upk, it adds j to CU, and it registers a new user with public key upk if $j \notin \mathsf{HU}$. Otherwise, it returns usk_j and sets $\mathsf{HU} \leftarrow \mathsf{HU} \setminus \{j\}$.
- $\mathcal{O}_{\mathsf{ObtIss}}(j, \mathbf{m})$: Given some $j \in \mathsf{HU}$ and attributes $\mathbf{m}$, it runs the protocol $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}_j, \mathsf{pp}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}_j, \mathsf{upk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$ assuming the roles of both O and user j. If successful, it increments the issuance counter ctr, stores the resulting credential and stores $(j, \mathsf{ctr}, \mathbf{m})$ in A. It returns $\top$ if the execution succeeded. If $j \notin \mathsf{HU}$, it simply returns $\perp$.
- $\mathcal{O}_{\mathsf{Obtain}}(j, \mathbf{m})$: Given 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.
- $\mathcal{O}_{\mathsf{Issue}}(j, \mathbf{m})$: Given j and attributes $\mathbf{m}$, it returns $\perp$ if $j \notin \mathsf{CU}$. Else, 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. If successful, it increments the issuance counter ctr, stores the credential and adds $(j, \mathsf{ctr}, \mathbf{m})$ in A.
- $\mathcal{O}_{\mathsf{Show}}(j', \mathbf{m}_{\mathcal{I}}^{(j')})$: It takes as input an issuance index j' and disclosed attributes $\mathbf{m}_{\mathcal{I}}^{(j')}$. The issuance index corresponds to a successfully issued credential $\mathsf{cred}^{(j')}$ on $\mathbf{m}_{\mathcal{I}}^{(j')}$ for a user j during the j'-th query to $\mathcal{O}_{\mathsf{IssObt}}$ or $\mathcal{O}_{\mathsf{Obtain}}$. If $j \in \mathsf{HU}$, it runs $\mathsf{Show}_{U,\mathcal{A}}((\mathsf{usk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}_{\mathcal{I}}^{(j')}, \mathsf{cred}^{(j')}, \mathcal{I}), \cdot)$ with the adversary posing as the verifier, and returns $\perp$ if $j \notin \mathsf{HU}$.

We expect the anonymous credentials system to be *correct*, i.e., honest executions of $\mathsf{Issue}_{O,U}$ succeed, and that honestly generated credentials pass verification in the $\mathsf{Show}_{U,V}$ protocol. We also want the anonymous credentials to be *anonymous*, that is showings should not reveal the credential nor the hidden attributes and user secret key, and different showings should be unlinkable. The security game is depicted in Figure 2.2. The scheme is *anonymous* if

$$|\mathbb{P}[b^* = b \wedge \mathcal{O}_{\mathsf{CU}} \text{ was not queried on } j_0 \text{ nor } j_1] - 1/2| \leq \mathsf{negl}(\lambda)$$

in the anonymity game. As in [JRS23], we assume without loss of generality that $(\mathsf{opk}, \mathsf{osk})$ are honestly generated. To ensure this in practice, the organization could provide a zero-knowledge proof of such a statement.

The *unforgeability* property not only captures the inability of an adversary to forge credential but more generally its inability to convince a verifier that they hold a valid credential. It therefore encompasses forgeries where an adversary would (1) impersonate an honest user, (2) trick the verifier with a falsified proof, and (3) forge a fresh credential, i.e., signature. We give the unforgeability game in Figure 2.3. The adversary wins the game if the challenger does not abort and if the challenger's output of the execution of Show is 1. We say that the anonymous credentials system is *unforgeable* if for all PPT adversary $\mathcal{A}$, its probability of winning is negligible.

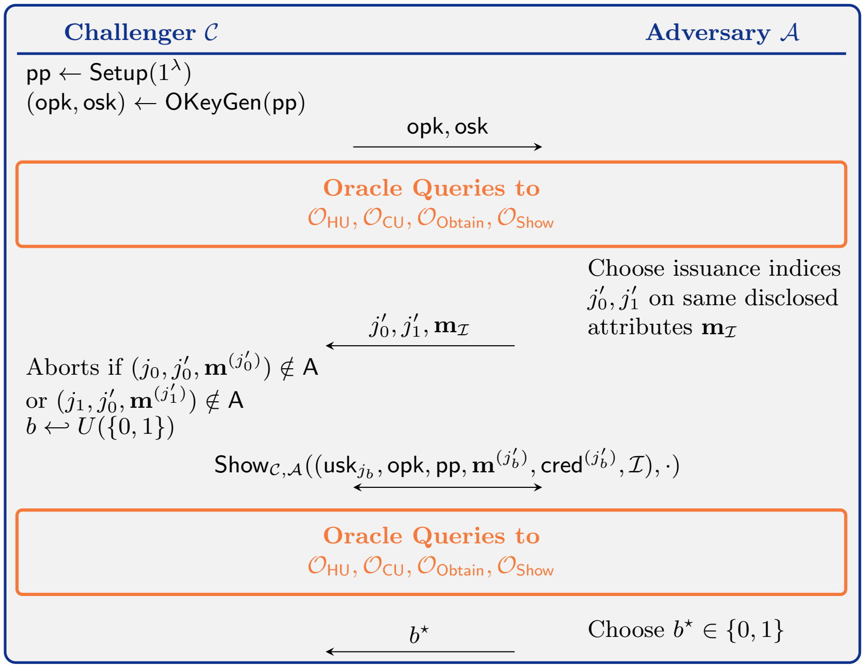
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$
$\mathsf{opk}, \mathsf{osk}$
Oracle Queries to
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Show}}$
Choose issuance indices j'_0, j'_1 on same disclosed attributes $\mathbf{m}_\mathcal{I}$
$j'_0, j'_1, \mathbf{m}_\mathcal{I}$
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
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Show}}$
Choose $b^\star \in \{0,1\}$
$b^\star$

Fig. 2.2. 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 $\mathbf{m}_\mathcal{I}^{(j'_\alpha)} = \mathbf{m}_\mathcal{I}$.

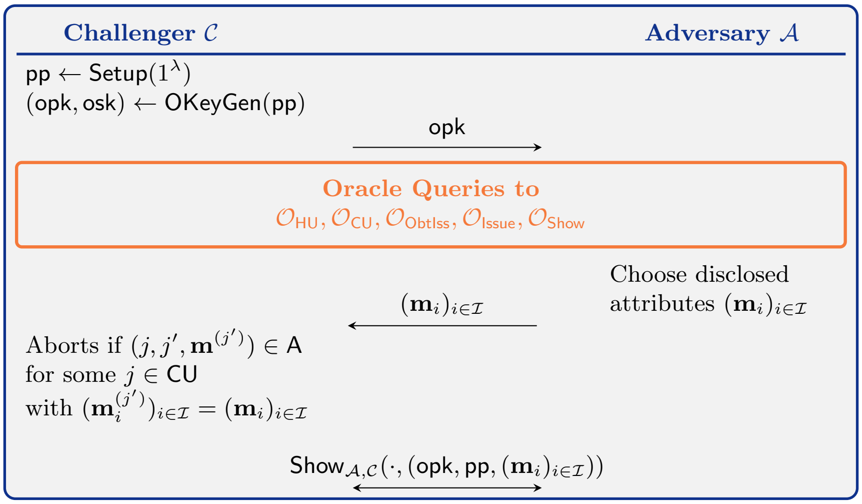
Challenger $\mathcal{C}$
Adversary $\mathcal{A}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$
opk
Oracle Queries to
$\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{ObtIss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}$
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. 2.3. Unforgeability Game for the Anonymous Credentials System.

Remark 2.2. The security model of [FHS19] assumes that all attributes, except the user's secret key, are revealed during issuance. Our scheme of Section 6 (like [JRS23]) is compatible with this model but also allows to conceal some attributes at issuance thanks to the zero-knowledge property of the proof system and the hiding property of the commitment scheme (M-LWE). More generally, our scheme enables selective disclosure of attributes at both the issuance and showing of credentials.

3 Preimage Sampler

We start by introducing the preimage sampler we use in our signature. We detail the overall loss of the sampler aiming for a finer parameter selection, and also to ease the floating-point precision analysis of Section 9. For that, we specify all the samplers used as subroutines (perturbation and gadget samplers).

3.1 Description

We here revisit the sampler introduced in [JHT22] which breaks the symmetry between the top and bottom parts of the perturbation $\mathbf{p}$ in [MP12], and use instead two different parameters s_1 and s_2. More precisely, we sample a perturbation over $R^{d(2+k)}$ of covariance

$$\begin{aligned}\mathbf{S} &= M_\tau\left(\begin{bmatrix} s_1^2\mathbf{I}_{2d} & \mathbf{0} \\ \mathbf{0} & s_2^2\mathbf{I}_{dk}\end{bmatrix} - s_{\mathbf{G}}^2\begin{bmatrix}\mathbf{R}\mathbf{R}^* & \mathbf{R} \\ \mathbf{R}^* & \mathbf{I}_{dk}\end{bmatrix}\right) \\ &= \begin{bmatrix} s_1^2\mathbf{I}_{2nd} & \mathbf{0} \\ \mathbf{0} & s_2^2\mathbf{I}_{ndk}\end{bmatrix} - s_{\mathbf{G}}^2\begin{bmatrix} M_\tau(\mathbf{R})M_\tau(\mathbf{R})^T & M_\tau(\mathbf{R}) \\ M_\tau(\mathbf{R})^T & \mathbf{I}_{ndk}\end{bmatrix},\end{aligned}$$

where s_2 will hopefully be much smaller than s_1 because $\mathbf{z}$ (as defined in Step 5 of Algorithm 3.2) has to be perturbed by a smaller amount than $\mathbf{Rz}$. For that, we use the module sampler from [BEP+21] (and Klein's sampler [Kle00,GPV08] for the gadget sampling part) which we describe in Algorithm 3.1. We slightly adapt their algorithm to our elliptic distribution featuring two Gaussian widths s_1 and s_2 instead of one. The only difference comes in step 3 in the definition of $\mathbf{S}'$ as the Schur complement is slightly different. The analysis still goes through in the very same way, as their sampler is an extension of the sampler from [GM18] which was already general enough to encompass the elliptic case.

Algorithm 3.1: SamplePerturb$(\mathbf{R}, s_1, s_2, s_{\mathbf{G}})$

Input: Trapdoor $\mathbf{R} \in R^{2d\times dk}$, Gaussian parameters $s_1, s_2, s_{\mathbf{G}} > 0$.

```
1. $\mathbf{p}_2 \hookleftarrow \mathcal{D}_{R^{dk},\sqrt{s_2^2 - s_{\mathbf{G}}^2}}$.
2. $\mathbf{c}_{2d} \leftarrow -s_{\mathbf{G}}^2/(s_2^2 - s_{\mathbf{G}}^2)\mathbf{R}\mathbf{p}_2$
3. $\mathbf{S}_{2d} \leftarrow s_1^2\mathbf{I}_{2d} - (s_{\mathbf{G}}^{-2} - s_2^{-2})^{-1}\mathbf{R}\mathbf{R}^*$.
4. for $i = 2d, \ldots, 1$ do
5.     Write $\mathbf{S}_i, \mathbf{c}_i$ as $\mathbf{S}_i = \begin{bmatrix}\mathbf{S}_i' & \mathbf{s}_i \\ \mathbf{s}_i^* & f_i\end{bmatrix}$ and $\mathbf{c}_i = \begin{bmatrix}\mathbf{c}_i' \\ d_i\end{bmatrix}$.
6.     $p_i \hookleftarrow \mathcal{D}_{R,\sqrt{M_\tau(f_i)},d_i}$.                ▷ SampleFz in [GM18, Fig. 4]
```

```
        $\mathbf{c}_{i-1} \leftarrow \mathbf{c}'_i + f_i^{-1}(p_i - d_i)\mathbf{s}_i$.
        $\mathbf{S}_{i-1} \leftarrow \mathbf{S}'_i - f_i^{-1}\mathbf{s}_i\mathbf{s}_i^*$.
    $\mathbf{p}_1 \leftarrow [p_1|\dots|p_{2d}]^T$.
Output: $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2)$        ▷ Statistically close to $\mathcal{D}_{R^{d(2+k)},\sqrt{\mathbf{S}}}$.
```

The first step only involves spherical sampling over $\mathbb{Z}^{ndk}$, while the sampling of p_i has a covariance $\sqrt{M_\tau(f)}$ for some $f \in K_\mathbb{R}$ verifying $f^* = f$. This can be handled using the sampler SampleFz from [GM18, Fig. 4] as is done in [BEP$^+$21]. We then obtain the following elliptic preimage sampler.

```
Algorithm 3.2: SamplePre($\mathbf{R}$; $\mathbf{A}'$, $\mathbf{u}$, $\mathbf{T}$, $s_1$, $s_2$, $s_\mathbf{G}$)

Input: Trapdoor $\mathbf{R} \in R^{2d \times dk}$, Matrix $\mathbf{A}' \in R_q^{d \times d}$, Syndrome $\mathbf{u} \in R_q^d$, Gaussian parameters $s_1, s_2, s_\mathbf{G} > 0$, tag $\mathbf{T} \in GL_d(R_q)$.

    $\mathbf{p} = [\mathbf{p}_1^T|\mathbf{p}_2^T]^T \hookleftarrow \mathcal{D}_{R^{d(2+k)},\sqrt{\mathbf{S}}}$.    ▷ SamplePerturb (Algorithm 3.1)
    $\mathbf{w} \leftarrow \mathbf{T}^{-1}(\mathbf{u} - [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{p}) \bmod qR$.    ▷ Syndrome correction
    $\mathbf{c} \leftarrow \mathbf{G}^{-1}(\mathbf{w})$.    ▷ Arbitrary solution
    $\mathbf{y} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{G}),s_\mathbf{G},-\mathbf{c}}$.    ▷ Klein (Algorithm 9.1)
    $\mathbf{z} \leftarrow \mathbf{c} + \mathbf{y}$.
    $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}$.
    $\mathbf{v}_2 \leftarrow \mathbf{p}_2 + \mathbf{z}$.

Output: $(\mathbf{v}_1, \mathbf{v}_2)$.
```

3.2 Security Analysis

We now study the closeness between the distribution outputted by Algorithm 3.2 and the ideal distribution. For that, we not only need to account for the loss captured in [MP12] but also from that of the samplers we use for the perturbation $\mathbf{p}$ and the gadget sample $\mathbf{y}$. These can be obtained from the literature. For the perturbation sampler of Algorithm 3.1, we follow the proof from [BEP$^+$21] but by specifying the loss at each step (as is done in [GM18] from which it takes inspiration). In the end, we obtain the following lemma.

Lemma 3.1 ([GM18, Thm. 4.1] adapted). *Let $\varepsilon \in (0, 1)$ be such that $\mathbf{S} - \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2\mathbf{I}_{nd(2+k)}$ is positive semi-definite. Denote by $\mathcal{P}$ the distribution outputted by* SamplePerturb*. Then, it holds that*

$$\forall \mathbf{p} \in R^{d(2+k)}, \mathcal{P}(\mathbf{p}) \in [\delta^{-1}, \delta] \cdot \mathcal{D}_{R^{d(2+k)},\sqrt{\mathbf{S}}}(\mathbf{p}),$$

where $\delta = ((1+\varepsilon)/(1-\varepsilon))^{6d(n-1)+1} \underset{\varepsilon \to 0}{\sim} 1 + 2(6d(n-1)+1)\varepsilon$.

For the gadget sampling step, we use Klein's sampler [Kle00,GPV08] which was thoroughly analyzed by Prest [Pre17].

Lemma 3.2 ([Pre17, Lem. 8] adapted). *Let $\varepsilon \in (0, 1/4)$ and let $s_\mathbf{G} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$. Denote by $\mathcal{P}$ the distribution outputted by Klein's sampler for* *the lattice* $\mathcal{L}_q^{\perp}(\mathbf{G})$ *and a center* $\mathbf{c}$*. Then, it holds that*

$$\forall \mathbf{y} \in R^{dk}, \mathcal{P}(\mathbf{y}) \in [\delta^{-1}, \delta] \cdot \mathcal{D}_{\mathcal{L}_q^{\perp}(\mathbf{G}), s_{\mathbf{G}}, \mathbf{c}}(\mathbf{y}),$$

where $\delta = ((1+\varepsilon/ndk)/(1-\varepsilon/ndk))^{ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2\varepsilon.$

Following the proof of [MP12, Thm. 5.5] but by using the imperfect samplers, we can derive the overall loss of SamplePre compared to the ideal distribution, i.e., $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}([\mathbf{A}|\mathbf{t}\mathbf{G}-\mathbf{A}\mathbf{R}]), \mathrm{diag}(s_1\mathbf{I}_{2nd}, s_2\mathbf{I}_{ndk})}$. We then obtain the following result.

Lemma 3.3 ([MP12, Thm. 5.5] adapted). *Let* $\varepsilon \in (0, 1/4)$ *and* $s_{\mathbf{G}} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$*. Assume that* $\mathbf{S} - \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2\mathbf{I}_{nd(2+k)}$ *and* $\mathbf{S} - s_{\mathbf{G}}^2/(s_{\mathbf{G}}^2 - 1)[M_\tau(\mathbf{R})^T|\mathbf{I}]^T[M_\tau(\mathbf{R})^T|\mathbf{I}]$ *are positive semi-definite. Denote by* $\mathcal{P}$ *the distribution outputted by* SamplePre*. Then, it holds that for all* $\mathbf{v} \in \mathcal{L}_q^{\mathbf{u}}([\mathbf{A}|\mathbf{T}\mathbf{G}-\mathbf{A}\mathbf{R}])$

$$\mathcal{P}(\mathbf{v}) \in [\delta_1, \delta_2] \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}([\mathbf{A}|\mathbf{T}\mathbf{G}-\mathbf{A}\mathbf{R}]), \mathrm{diag}(s_1\mathbf{I}_{2nd}, s_2\mathbf{I}_{ndk})}(\mathbf{v}),$$

where

$$\delta_1 = \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{6d(n-1)+3}\left(\frac{1-\varepsilon/ndk}{1+\varepsilon/ndk}\right)^{ndk} \underset{\varepsilon\to 0}{\sim} 1 - 2(6d(n-1)+4)\varepsilon$$

$$\delta_2 = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{6d(n-1)+2}\left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2(6d(n-1)+3)\varepsilon.$$

3.3 Parameter Setting

To guarantee that the sampler is correct, we need to satisfy parameter constraints of Lemma 3.3. For that, we set $s_{\mathbf{G}} = \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$ and then determine the values of s_1, s_2 so that $\mathbf{S}$ verifies the necessary conditions. We thus use the following lemma.

Lemma 3.4. *Let* m, ℓ *be positive integers,* $\mathbf{R} \in \mathbb{R}^{m\times\ell}$*, and* α, β, γ *positive reals. The matrix*

$$\mathbf{S} = \begin{bmatrix} \alpha^2\mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \beta^2\mathbf{I}_\ell \end{bmatrix} - \gamma^2 \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_\ell \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & \mathbf{I}_\ell \end{bmatrix}$$

is positive definite if and only if $\alpha > \sqrt{1 + 1/(c^2-1)}\gamma\|\mathbf{R}\|_2$ *and* $\beta > c\gamma$ *for some* $c > 1$*. For* $c = \sqrt{2}$ *it yields* $\alpha > \sqrt{2}\gamma\|\mathbf{R}\|_2$ *and* $\beta > \sqrt{2}\gamma$*.*

Proof. We can re-write $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} \alpha^2\mathbf{I}_m - \gamma^2\mathbf{R}\mathbf{R}^T & -\gamma^2\mathbf{R} \\ -\gamma^2\mathbf{R}^T & (\beta^2-\gamma^2)\mathbf{I}_\ell \end{bmatrix} =: \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}.$$

Then, by using the characterization by Schur complements, it holds that $\mathbf{S}$ is positive definite if and only if $\mathbf{C}$ is positive definite and $\mathbf{S}/\mathbf{C} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$ is also positive definite. This means having

$$(\beta^2-\gamma^2)\mathbf{I}_\ell \quad \text{and} \quad (\alpha^2\mathbf{I}_m - \gamma^2\mathbf{R}\mathbf{R}^T) - (-\gamma^2)\mathbf{R}\cdot(\beta^2-\gamma^2)^{-1}\mathbf{I}_\ell\cdot(-\gamma^2)\mathbf{R}^T$$

positive definite. The condition translates to $\beta > \gamma$ and $\alpha^2 > \lambda_{\max}((\gamma^2 + \gamma^4(\beta^2 - \gamma^2)^{-1})\mathbf{R}\mathbf{R}^T)$, where $\lambda_{\max}$ denotes the largest eigenvalue. It comes down to $\beta > c\gamma$ and $\alpha > \sqrt{1 + 1/(c^2-1)}\gamma\|\mathbf{R}\|_2$ for any $c > 1$ as claimed. □

As a result, we have to choose the Gaussian widths s_1 and s_2 such that $\sqrt{s_1^2 - \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2} \geq \sqrt{2}s_{\mathbf{G}}\|\mathbf{R}\|_2$ and $\sqrt{s_2^2 - \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2} \geq \sqrt{2}s_{\mathbf{G}}$, and also such that $s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2 - 1)}\|\mathbf{R}\|_2$ and $s_2 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}$. We can therefore set $s_1 = \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\|\mathbf{R}\|_2$ and $s_2 = \sqrt{2s_{\mathbf{G}}^2 + \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2}$ and still inherit from the analysis of [MP12]. This allows us to drastically reduce the size of the bottom part for free, while keeping the size of the top part (almost) the same as before. Additionally, the overall norm of $\mathbf{v}$ is smaller which can result in slightly increased concrete security. Using the perturbation sampler of Algorithm 3.1 leads to slightly improved parameters over [JHT22], but, more importantly, a drastic computational efficiency gain over the Peikert sampler [Pei10] implicitly used in [MP12] and in turn [JHT22]. We also note that [JHT22] only provide the simulation for uniform targets $\mathbf{u}$, and not arbitrary ones as in [MP12] and our case.

4 Trapdoor Switching

We now formalize the (partial) trapdoor switching sketched in Section 1 and the detailed loss it incurs in the following lemma. We give a more detailed explanation in Section 5.1 on how it is used in the security proof of our signature.

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

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

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

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

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

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

Proof. We additionally define $\overline{\mathbf{A}}' = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]$ and refer to $\mathbf{v}$ as the random vector $[\mathbf{v}_1^T|\mathbf{v}_2^T|\mathbf{v}_3^T]^T$. First starting from $\mathcal{P}_1$, conditioned on $\mathbf{v}_3 = \overline{\mathbf{v}_3}$, Lemma 3.3 yields that the distribution of $(\mathbf{v}_1, \mathbf{v}_2)$ is $[\delta_1, \delta_2]$-close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}_3}(\overline{\mathbf{A}}'),\mathrm{diag}(s_1,s_2)}$ where $\mathbf{u}_3 = \mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1})\overline{\mathbf{v}_3} \bmod qR$. However, this distribution corresponds exactly to the distribution $\mathcal{D}_{R^{d(2+k)},\mathrm{diag}(s_1,s_2)}$ conditioned to $\overline{\mathbf{A}}'[\mathbf{v}_1^T|\mathbf{v}_2^T]^T = \mathbf{u}_3 \bmod qR$. Hence, we have

$$\begin{aligned}
&\mathcal{P}_1(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) \in [\delta_1, \delta_2] \cdot \mathcal{D}_{R^k,s_2}(\overline{\mathbf{v}_3})\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}_3}(\overline{\mathbf{A}}'),\mathrm{diag}(s_1,s_2)}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}) \\
&= [\delta_1, \delta_2] \cdot \left(\mathcal{D}_{R^{d(2+k)+k},\mathrm{diag}(s_1,s_2,s_2)} \middle| \overline{\mathbf{A}}\mathbf{v} = \mathbf{u} \bmod qR\right)(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) \\
&= [\delta_1, \delta_2] \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}),\mathrm{diag}(s_1,s_2,s_2)}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}).
\end{aligned}$$

Similarly, starting from $\mathcal{P}_2$ and conditioning on $\mathbf{v}_{2,i} = \overline{\mathbf{v}_{2,i}}$ yields

$$\mathcal{P}_2(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) \in [\delta_1, \delta_2] \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}),\mathrm{diag}(s_1,s_2,s_2)}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}).$$

Combining both gives the result with a loss $\delta = \delta_2/\delta_1$. The expression of δ_1, δ_2 and their asymptotic equivalent for small ε are obtained from Lemma 3.3 and yield the expression and asymptotic equivalent for δ. □

Compared to a version of our scheme with a full second trapdoor slot, that is with $\mathbf{A}_3$ having kd columns, our trapdoor switching technique provides significant gains. More concretely, it reduces the size of the signature from 9.90 KB to 6.81 KB, i.e., a 32% improvement. Most importantly, it also reduces the dimensionality of the witness vector in the credential showing proof. As a result, the proof size goes from 93.47 KB down to 79.58 KB, i.e. a 15% gain. We note that our partial trapdoor technique can be used in other constructions as well. For example, the group signature [LNPS21] would benefit from our technique allowing to reduce the group signature size as well as the user secret key size.

5 The Signature

5.1 Intuition

We here provide the intuition behind the most noticeable modifications we are making to [JRS23]. Other modifications, such as finer precision analysis or bound optimisations are not discussed here because they do not intrinsically change the construction of [JRS23], but primarily because they require to be presented with many intricate details that do not fit this section. Those details are thus postponed to Sections 5.3, 3 and 9.

Step 0: The [JRS23] construction. In [JRS23], a signature $\mathbf{v}$ on a message $\mathbf{m}$ is a vector of dimension roughly $2dk$ following a spherical Gaussian distribution such that

$$\mathbf{A}_{\mathbf{T}}\mathbf{v} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{v} = \mathbf{u} + \mathbf{Ar} + \mathbf{Dm} \bmod qR.$$

We refer to Section 1 for a definition of each of the elements involved in this relation. As explained in the same section, moving from the statistical setting to the (more efficient) computational setting introduces a problem in the security proof which is currently solved (see e.g., [dPLS18,LNPS21]) by adding a second trapdoor. In the case of MP trapdoors [MP12], we would end up with a matrix $\mathbf{A_T} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathbf{G} - \mathbf{AR}']$ of dimension $2(d + k)$.

Step 1: Reducing dimension of $\mathbf{A_T}$ (and hence $\mathbf{v}$). We recall that the core idea of security proofs for this family of signatures is to transform the public key $\mathbf{AR}$ into $\mathbf{T}^\star\mathbf{G} + \mathbf{AR}$, for some tag matrix $\mathbf{T}^\star$. This is currently done through hybrid games where $\mathbf{AR}$ is replaced at some stage by a random matrix, without associated trapdoor. The second trapdoor therefore takes over signature generation during this intermediate game before being discarded when the transformation to $\mathbf{T}^\star\mathbf{G} + \mathbf{AR}$ is complete.

We significantly improve over this approach by presenting a new strategy that does not need a full extra trapdoor, but only part of one. Using this extra partial trapdoor slot in a hybrid argument allows the same security proof (namely moving from $\mathbf{AR}$ to $\mathbf{T}^\star\mathbf{G} + \mathbf{AR}$) to go through in a much more compact way.

The idea is as follows. The gadget matrix $\mathbf{G} = \mathbf{I}_d \otimes \mathbf{g}^T$ can be written as $[\mathbf{e}_1 \otimes \mathbf{g}^T| \dots |\mathbf{e}_d \otimes \mathbf{g}^T]$, where $\mathbf{e}_j$ is the j-th canonical vector of $\mathbb{R}^d$. For clarity, we define $\mathbf{G}_j = \mathbf{e}_j \otimes \mathbf{g}^T \in R^{d\times k}$. Assume we have a matrix of the form $\overline{\mathbf{A}} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}]$ with $\mathbf{T} = \mathrm{diag}(\mathfrak{t}_1, \dots, \mathfrak{t}_d)$. To sample a preimage of $\mathbf{u}$ by $\overline{\mathbf{A}}$, we could proceed in two ways that we show are statistically close (see Section 4). The first way is to sample $\mathbf{v}_3$ from $\mathcal{D}_{R^k,s_2}$ and then use $\mathsf{SamplePre}$ with the trapdoor $\mathbf{R}$ to find a preimage of $\mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1})\mathbf{v}_3$. The second way is to essentially exchange the j-th block of $\mathbf{TG} - \mathbf{AR}$, that is, the columns $jk+1, \dots, (j+1)k$, by the final block $\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}$. Concretely, one samples $\mathbf{v}_{2,j}$ from $\mathcal{D}_{R^k,s_2}$ and then samples $[\mathbf{v}_1, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,j-1}, \mathbf{v}_3, \mathbf{v}_{2,j+1}, \dots, \mathbf{v}_{2,d}]$ from $\mathsf{SamplePre}$ with the trapdoor $[\mathbf{R}_1| \dots |\mathbf{R}_{j-1}|\mathbf{R}_{d+1}|\mathbf{R}_{j+1}| \dots |\mathbf{R}_d]$ on the syndrome $\mathbf{u} - (\mathfrak{t}_j\mathbf{G}_j - \mathbf{AR}_j)\mathbf{v}_{2,j}$, and with tag $\mathrm{diag}(\mathfrak{t}_1, \dots, \mathfrak{t}_{j-1}, \mathfrak{t}_{d+1}, \mathfrak{t}_{j+1}, \dots, \mathfrak{t}_d)$. The point of this second case is that $\mathbf{R}_j$ is no longer needed for preimage sampling, so the unused block $\mathbf{AR}_j$ can be replaced in the public key by k random vectors without impacting the ability to answer signature queries. Those random vectors can then be replaced by $\mathfrak{t}'_j\mathbf{G}_j + \mathbf{AR}_j$ for arbitrary $\mathfrak{t}'_j$ in a second game hop. In both cases, indistinguishability between those games relies on the M-LWE assumption.

A standard hybrid argument is then used to finalize the proof. More concretely, we set $\mathfrak{t}_{d+1} = 1$. At the beginning of the proof, any signature with tag $\mathfrak{t}$ is answered normally with $\mathfrak{t}_i = \mathfrak{t}$ for $i \leq d$. Then, at the j-th hybrid, $j - 1$ applications of the strategy above lead to a situation where the first $(j - 1)$ blocks $\mathbf{AR}_i$ of the public key has been replaced by $\mathfrak{t}^+\mathbf{G}_i + \mathbf{AR}_i$, which means that signatures with tag $\mathfrak{t}$ are actually generated using $\mathfrak{t}_1 = \ldots = \mathfrak{t}_{j-1} = \mathfrak{t} - \mathfrak{t}^+$ and $\mathfrak{t}_i = \ldots = \mathfrak{t}_d = \mathfrak{t}$. Note that this is transparent to the adversary as the signatures do not leak any information on the actual tag: this is actually the core argument of the security proofs in [MP12,JRS23] where the adversary obliviously use tags $(\mathbf{T} - \mathbf{T}^\star)$. Moreover, generating a signature for tag $\mathfrak{t}^+$ is still possible using the very classical approach consisting in programming the public key $\mathbf{u}$ accordingly. Our reduction can thus answer all signing queries at any stage using only this extra block $\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}$. As a consequence, the dimension of $\mathbf{A_T}$ and hence the one of our signatures will be $2d + k(d+1)$ instead of $2(d + kd)$, which leads to much better performance. This idea of trapdoor switching is formalized in Section 4, and the full security proofs using it can be found in Section 5.3.

Step 2: Changing signature distribution. At this stage, we then have a shorter signature $\mathbf{v}$ which still follows a spherical Gaussian distribution. The main step in the procedure to generate $\mathbf{v}$ is the preimage sampling algorithm used on syndromes of the form $\mathbf{u} + \mathbf{A}\mathbf{r} + \mathbf{D}\mathbf{m} \bmod qR$. When using the MP sampler, as is done in [JRS23], the preimage must be perturbed so as to hide information on the trapdoors. Concretely, this is done by generating a perturbation $\mathbf{p}$ which is then added to the preimage whereas the syndrome is modified accordingly. In [MP12], all parts of $\mathbf{p}$ were generated so as to obtain the spherical distribution mentioned above, which is not optimal as noted in, e.g., [JRS23]. One could consider alternatives such as those described in [LW15,CGM19] but their analysis requires statistically uniform syndromes, not computationally uniform ones, which would bring us back to Step 0.

To find a middle way between the original MP sampler and the ones requiring statistical regularity arguments, e.g., [LW15,CGM19], we revisit the sampler which was very recently introduced in [JHT22]. The idea is to tweak the MP sampler to generate the perturbation $\mathbf{p}$ with two different Gaussian widths, one for the upper part and one for the lower part, leading to elliptical Gaussians. This allows for reducing the size of preimages, and thus signatures, at no cost on security. We provide details on this elliptical sampler in Section 3.

Step 3: Removing Signer's Randomness. As explained above, the goal of the security proofs of signature schemes based on the MP sampler is to end up with a situation where the reduction can normally answer all signing queries but one, for which it has no trapdoor. For this special query, the reduction leverages some information hidden in the public parameters but, as the latter are defined at the beginning of the game, they do not necessarily compensate the $\mathbf{D}\mathbf{m}$ component of the syndrome which is adaptively chosen by the adversary. As a consequence, the distribution of the signature in this case may not be correct, leading the reduction to fail. In [JRS23], this problem is solved using a rather conventional approach where the signer contributes to the syndrome. Concretely, instead of computing a preimage of $\mathbf{u} + \mathbf{D}\mathbf{m} \bmod qR$, it selects a random vector $\mathbf{r}$ and then computes a preimage of $\mathbf{u} + \mathbf{D}\mathbf{m} + \mathbf{A}\mathbf{r} \bmod qR$. This additional randomness $\mathbf{r}$, chosen with the knowledge of $\mathbf{m}$, is sufficient to prove security using a standard noise drowning argument with the Rényi divergence as shown in [JRS23].

Besides making the signing procedure more complex, the downside of this approach is that it adds an element $\mathbf{r}$ to the signature. Although it can be merged with the signature thanks to the approach of [JRS23], it still has a cost as it increases the norm of the first part of the signature. To remove $\mathbf{r}$, we follow in our proof a different approach based on rejection sampling, as in [CKLL19]. The core idea is to abort the reduction if the message $\mathbf{m}$ leads to an invalid distribution of the signature while tolerating a small amount of leakage using Lemma 2.9 (contrarily to [CKLL19]) so as to improve performance. As this leakage only occurs once in the reduction, it does not significantly impact security. In all cases, this approach only entails a small constant reduction loss factor compared to the previous one based on the Rényi divergence. More precisely, we achieve a constant loss factor, but without having to increase the Gaussian width by a $\Theta(\sqrt{\lambda})$ factor. Decreasing the reduction loss allows us to use much smaller parameters as we need to aim for around 165 bits of M-SIS core-SVP hardness instead of 210 in [JRS23].

We nevertheless stress that this only allows to remove the signer's randomness. Some situations (e.g. obtaining a signature in a privacy-preserving protocol) may indeed require the user to hide his message by adding $\mathbf{A}\mathbf{r}_u$ to the commitment $\mathbf{D}\mathbf{m}$ and this remains true in our case. We will therefore need to consider two variants of our scheme, one for the standalone version of our signature and one for usage in the situations mentioned above. Actually, the only difference will be located in the verification bound on the first part of the signature ($\mathbf{v}_1$). For the signature itself, we have $\|\mathbf{v}_1\|_2 \leq B_1$ where B_1 is determined by Lemma 2.7, while in the protocols, we have $\|\mathbf{v}_1\|_2 \leq B_1 + \|\mathbf{r}_u\|_2$. At this point, changing to a rejection-based analysis also improves upon [JRS23]. The choice of Gaussian randomness seemed motivated by the use of the Rényi divergence in the noise drowning step. Using a rejection-based method allows us to rely on a computationally hiding commitment and use $\mathbf{r}_u$ to be composed of binary polynomials, which results in only a $\sqrt{2nd}$ additive term in the verification bound.

In the end, our signature $\mathbf{v}$ on a message $\mathbf{m}$ is now a vector of dimension $2d + k(d+1)$ following an elliptical distribution such that

$$\mathbf{A}_{\mathbf{T}}\mathbf{v} = [\mathbf{A}|\mathbf{T}\mathbf{G} - \mathbf{A}\mathbf{R}|\mathbf{A}_3]\mathbf{v} = \mathbf{u} + \mathbf{D}\mathbf{m} \bmod qR,$$

where $\mathbf{A}_3$ is a $d \times k$ random matrix. Together with the optimizations mentioned at the beginning of this section, it leads to significant performance improvements, as illustrated in Tables 8.1 and 8.4.

5.2 The Scheme

The formal description of the Setup, Keygen, Sign and Verify algorithms that constitute our signature scheme is provided below.

Algorithm 5.1: Setup

```
Input: Security parameter $\lambda$.

Choose a positive integer $d$.
Choose $\kappa \leq n$ to be a power of two.
Choose a prime $q$ s.t. $q = 2\kappa + 1 \bmod 4\kappa$ and $q \geq (2\sqrt{\kappa})^{\kappa}$.
Choose positive integer $w$.                                   ▷ Hamming weight of tags
Choose positive integer $b$.                                   ▷ Gadget base
$\mathcal{T}_w \leftarrow \{\mathsf{t} \in \mathcal{T}_1 : \|\mathsf{t}\|_1 = w\}$.       ▷ Tag space
$k \leftarrow \lceil \log_b q \rceil$.
Choose a positive integer $m$.                     ▷ Maximum bit-size of $\mathbf{m}$ is $nm$
$\mathbf{G} = \mathbf{I}_d \otimes [1 \cdots b^{k-1}] \in R_q^{d \times dk}$.                 ▷ Gadget matrix
$r \leftarrow \sqrt{\ln(2nd(2+k)(1+\varepsilon^{-1}))/\pi}$.     ▷ $r \gtrsim \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})$
```

```
11. $s_{\mathbf{G}} \leftarrow r\sqrt{b^2+1}$.                                          ▷ Gadget sampling width
12. $s_1 \leftarrow \max\left(\sqrt{\frac{\pi}{\ln(2)}}n\sqrt{dm}, \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2-1}} \cdot \frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)\right)$.      ▷ Top preimage width
13. $s_2 \leftarrow r\sqrt{2b^2+3}$.                                         ▷ Bottom preimage width
14. $\alpha \leftarrow \frac{s_1}{n\sqrt{dm}}$                                        ▷ Rejection sampling slack
15. $M \leftarrow \exp(\pi/\alpha^2)$.                                        ▷ Repetition rate
16. $\mathbf{D} \hookleftarrow U(R_q^{d \times m})$.                                     ▷ Message Commitment Key
17. $\mathbf{A}' \hookleftarrow U(R_q^{d \times d})$.
18. $\mathbf{A}_3 \hookleftarrow U(R_q^{d \times k})$.
19. $\mathbf{u} \hookleftarrow U(R_q^{d})$.
20. $\mathbf{A} \leftarrow [\mathbf{I}_d|\mathbf{A}'] \in R_q^{d \times 2d}$.
```

Output: $\mathsf{pp} = (\lambda, n, d, q, w, b, k, m, r, s_{\mathbf{G}}, s_1, s_2, \alpha, M, \mathbf{D}, \mathbf{A}', \mathbf{A}_3, \mathbf{u})$.

Algorithm 5.2: KeyGen

Input: Public parameters pp as in Algorithm 5.1.

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

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

Algorithm 5.3: Sign

Input: Signing key sk, Message $\mathbf{m} \in T_1^m$, Public key pk, Public Parameters pp, State st

```
1. $\mathbf{c} \leftarrow \mathbf{Dm} \bmod qR$.                                   ▷ Biding commitment to $\mathbf{m}$
2. $\mathsf{t} \leftarrow F(\mathsf{st})$.                                              ▷ $\mathsf{t} \in \mathcal{T}_w$
3. $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
4. $(\mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u}+\mathbf{c}-\mathbf{A}_3\mathbf{v}_3 \bmod qR, \mathsf{t}, s_1, s_2, s_{\mathbf{G}})$
5. if $\|\mathbf{v}_1\|_2 > B_1 \;\vee\; \|\mathbf{v}_2\|_2 > B_2 \;\vee\; \|\mathbf{v}_3\|_2 > B_3$   goto 3).
6. $\mathsf{st} \leftarrow \mathsf{st}+1$.
7. Parse $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T]^T$ with $\mathbf{v}_{1,1}, \mathbf{v}_{1,2} \in R^d$.
```

Output: $\mathsf{sig} = (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$.

Algorithm 5.4: Verify

Input: Public key pk, Message $\mathbf{m} \in T_1^m$, Signature sig, Public Parameters pp.

```
1. $\mathbf{v}_{1,1} \leftarrow \mathbf{u}+\mathbf{Dm}-\mathbf{A}'\mathbf{v}_{1,2}-(\mathsf{t}\mathbf{G}-\mathbf{B})\mathbf{v}_2-\mathbf{A}_3\mathbf{v}_3 \bmod qR \in R^d$.
2. $\mathbf{v}_1 \leftarrow [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T]^T$
3. $b_1 \leftarrow \|\mathbf{v}_1\|_2 \leq B_1$.                                   ▷ $B_1 = c_{2nd}s_1\sqrt{2nd}$
4. $b_2 \leftarrow \|\mathbf{v}_2\|_2 \leq B_2$.                                   ▷ $B_2 = c_{ndk}s_2\sqrt{ndk}$
5. $b_3 \leftarrow \|\mathbf{v}_3\|_2 \leq B_3$.                                   ▷ $B_3 = c_{nk}s_2\sqrt{nk}$
6. $b_4 \leftarrow \mathsf{t} \in \mathcal{T}_w$.
7. $b_5 \leftarrow \mathbf{m} \in T_1^m$.
```

Output: $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5$.   ▷ 1 if valid, 0 otherwise

Remark 5.1. As usual with tag-based signatures, a tag t must never be used twice. This condition is easily met when plugged in an anonymous credential system as one can derive the tag from, e.g., user-dependent information. But at this stage, we must remain general and thus choose to describe our signature in its stateful variant, as was done in [JRS23]. We refer to the latter paper for further discussion on this topic.

Lemma 5.1 (Correctness). *The signature scheme of Algorithms 5.1, 5.2, 5.3, and 5.4 is correct.*

Proof. Let pp, and $(\mathsf{pk}, \mathsf{sk}) = ((\mathbf{B}, \mathbf{u}), \mathbf{R})$ be obtained by running $\mathsf{Setup}(1^\lambda)$, and $\mathsf{KeyGen}(\mathsf{pp})$ respectively. Let $\mathbf{m} \in \{0,1\}^m$ be an arbitrary message and $(\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3) \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{m}, \mathsf{pk}, \mathsf{pp}, \mathsf{st})$ a signature. We define

$$\mathbf{v}_{1,1} = \mathbf{u} + \mathbf{D}\mathbf{m} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR \in R^d.$$

Then, $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$ and $\mathbf{v}_2$ were obtained from $\mathsf{SamplePre}(\mathbf{R}, \mathbf{A}, \mathsf{t}\mathbf{I}_d, \mathbf{u} + \mathbf{D}\mathbf{m} - \mathbf{A}_3\mathbf{v}_3 \bmod qR, s_1, s_2, s_{\mathbf{G}})$. Using the same argument as the one from the proof of Lemma 4.1, we get that the distribution of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is $[\delta_1, \delta_2]$-close from the elliptical distribution $\mathcal{D}_{R^{2d(1+k)}, \mathrm{diag}(s_1, s_2, s_2)}$ conditioned on $\mathbf{A}\mathbf{v}_1 + (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 = \mathbf{u} + \mathbf{D}\mathbf{m} \bmod qR$, where δ_1, δ_2 are defined in Lemma 3.3. Applying Lemma 2.7 yields the bounds $B_1 = c_{2nd} s_1 \sqrt{2nd}$, $B_2 = c_{ndk} s_2 \sqrt{ndk}$ and $B_3 = c_{nk} s_2 \sqrt{nk}$ on $\|\mathbf{v}_1\|_2, \|\mathbf{v}_2\|_2, \|\mathbf{v}_3\|_2$. It gives that $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5 = 1$ except with probability $\delta_2 2^{-(\lambda + O(1))}$ by definition of c_{2nd}, c_{nkd}, c_{nk}. Since we set ε so that $\delta_2 = 1 + O(1)$, we indeed obtain the correctness with overwhelming probability as claimed. Note that since we reject signatures that exceed the bounds during the signing process, the correctness of outputted signatures is actually guaranteed. Nevertheless, the correctness error we just derived is helpful to establish that generated signatures are never rejected except with negligible probability, thus bounding the number of rejections during the signing procedure. □

Remark 5.2. We note that the bounds B_i are set so that generated signatures are rejected only with negligible probability $2^{-(\lambda + O(1))}$, by definition of c_{2nd}, c_{nkd}, and c_{nk}. We can have smaller tailcuts by aiming for a probability bound of say 2^{-12} so that all three bounds are verified except with probability at most 2^{-10}. This would slightly improve the signature sizes and the M-SIS bounds used in the security assessment, but at the expense of rejecting signatures more often. It then provides a trade-off between size performance and computational performance. We decide not to feature this optimization for clarity of presentation, and because it is already quite standard in lattice schemes parameter optimization, e.g., [PFH+20].

5.3 Security Analysis.

We now give the formal security statement of our signature scheme. We distinguish between two different types a forgeries and treat them separately. Combining both Theorem 5.1 and 5.2 proves the EUF-CMA security.

Theorem 5.1. *An adversary produces a forgery* $(\mathsf{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ *of* Type ❶ *if the tag* $\mathsf{t}^\star$ *does not collide with the tags of the signing queries. The advantage of any* PPT *adversary* $\mathcal{A}$ *in producing a type* ❶ *forgery is at most*

$$\mathsf{Adv}_{\text{❶}}[\mathcal{A}] \lesssim h^{\circ d}(C(|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}})$$

where C *is a small constant from Lemma 2.4,* $\varepsilon_{\text{M-SIS}}$ *is the hardness bound of* $\text{M-SIS}_{n,d,2d+k+m+1,q,\beta_{\text{❶}}}$ *for*

$$\beta_{\text{❶}} = \sqrt{(B_1 + \sqrt{nd}B_2)^2 + B_3^2 + nm + 1},$$

and $h^{\circ d}$ *is the function* h *composed* d *times, where* h *is defined by*

$$h(x) = k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(2k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(k\varepsilon_{\text{M-LWE}} + x\right)^{1-1/2\lambda}\right)^{1-1/2\lambda},$$

with $\varepsilon_{\text{M-LWE}}$ *the hardness bound of* $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$.

Proof. Throughout the proof, we consider an PPT adversary $\mathcal{A}$ interacting with the challenger $\mathcal{B}$, and which aims at producing a valid Type ❶ forgery. We proceed by a game hop to modify the distribution of the view of $\mathcal{A}$ in a way that is indistinguishable from the real distribution. In the last game, the constructed elements that compose the distribution given to $\mathcal{A}$ allow to easily exploit the forgery to obtain a solution to M-SIS. Under the assumption that M-SIS is hard, it should thus be infeasible for $\mathcal{A}$ to produce a valid type ❶ forgery. We proceed using a game-based proof which follows the sequence summarized in Figure 5.1.

```
– $G_0$                                                   ▷ Original game
– $G_1$                                                   ▷ Sample tags at the start
– For $j \in [d]$
–       For $i \in [0, 9]$                                ▷ Hiding tag guess in partial key $j$
–             $G_{j,i}$
– Solve M-SIS using $\mathcal{A}$ against $G_{d,9}$.
```

Fig. 5.1. Overview of the unforgeability reduction (type ❶)

Games Hops. We define the following games which are composed of three stages: setup, queries, forgery. Past the queries stage, the view of the adversary does not change so we only describe the first two stages. The matrix $\mathbf{A}'$ (and in turn $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$), the matrix $\mathbf{D}$, and the syndrome $\mathbf{u}$ are always generated the same way, i.e., $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$, $\mathbf{D} \hookleftarrow U(R_q^{d\times m})$ and $\mathbf{u} \hookleftarrow U(R_q^d)$, and we thus do not specify them in the games below. In each game, the view of $\mathcal{A}$ is $(\mathbf{A}, \mathbf{D}, \mathbf{B}, \mathbf{u}, \mathbf{A}_3, (\mathsf{sig}^{(i)})_{i\in[Q]})$.

<ins>Game G_0.</ins> This corresponds to the original unforgeability game where the key material generation and signing queries are handled honestly. More precisely, we have

G_0

Setup

1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d \times dk}$ s.t. $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
2. $\mathbf{B} = \mathbf{AR} \bmod qR$
3. $\mathbf{A}_3 \hookleftarrow U(R_q^{d \times k})$

Queries

Given $\mathbf{m}^{(i)} \in T_1^m$, compute $\mathfrak{t}^{(i)} = F(\mathsf{st})$ and increment st. Then

1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k, s_2}$
2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathfrak{t}^{(i)}, s_1, s_2, s_{\mathbf{G}})$
3. Send $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.

Game G_1. In G_1, we simply change the way tags are generated. Instead of computing $\mathfrak{t}^{(i)}$ at each signing query, we first generate and store all the Q tags during the setup stage. In the query stage, we simply look-up the corresponding tag. It also samples a tag guess $\mathfrak{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathfrak{t}^{(i)}; i \in [Q]\})$, but it is so far not used. The view is exactly the same in G_1 because we only changed the moment when the tags are generated. Since they are generated deterministically from the state, both views are identically distributed.

We are now aiming to hide the tag guess within the public key, that is replace the public key $\mathbf{B} = \mathbf{AR}$ by $\mathbf{B} = \mathbf{AR} + \mathfrak{t}^+\mathbf{G}$, while keeping the ability to answer signing queries. For that, we proceed with a hybrid argument defined by a sequence of games $G_{j,\ell}$ for $j \in [d]$ and $\ell \in [0, 9]$. Recall the notation $\mathbf{G}_i = \mathbf{e}_i \otimes \mathbf{g}^T$ from Section 4, which corresponds to having the gadget only on the i-th row, thus allowing to invert only to i-th entry of a syndrome. In game $G_{j,9}$, the public key has been transformed to $\mathbf{B} = \mathbf{AR} + [\mathfrak{t}^+\mathbf{G}_1| \ldots |\mathfrak{t}^+\mathbf{G}_j|\mathbf{0}| \ldots |\mathbf{0}]$. We construct the games so that $G_{1,0} = G_1$, that for all $j \in [d-1]$, $G_{j,9} = G_{j+1,0}$ and we give detailed arguments to go from $G_{j,0}$ to $G_{j,9}$. Let $j \in [d]$.

Game $G_{j,0}$. In this game, the challenger performs the setup phase as follows. It computes all the $\mathfrak{t}^{(i)}$ at the outset and samples a tag guess $\mathfrak{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathfrak{t}^{(i)}; i \in [Q]\})$. It then samples $(\mathbf{R}_i)_{i \in [d]}$ from $\mathcal{B}_1^{2d \times k}$ such that $\mathbf{R} = [\mathbf{R}_1| \ldots |\mathbf{R}_d]$ satisfies $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$. Then, for $i \in [j-1]$ it defines $\mathbf{B}_i = \mathbf{AR}_i + \mathfrak{t}^+\mathbf{G}_i \bmod qR$, and for $i \in [j, d]$ it defines $\mathbf{B}_i = \mathbf{AR}_i \bmod qR$. It then constructs $\mathbf{B} = [\mathbf{B}_1| \ldots |\mathbf{B}_d]$ as the public key. Note that when $j = 1$ we simply have $\mathbf{B} = \mathbf{AR} \bmod qR$. It then samples $\mathbf{A}_3$ from $U(R_q^{d \times k})$, and sends the public key and public parameters to $\mathcal{A}$.

When receiving a signing query on $\mathbf{m}^{(i)}$, the challenger looks-up the tag $\mathfrak{t}^{(i)}$ and proceeds as follows. It samples $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k, s_2}$, and then samples

$$(\mathbf{v}_1^{(i)}, \mathbf{v}_2^{(i)}) = \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathbf{T}_j, s_1, s_2, s_{\mathbf{G}}),$$

where

$$\mathbf{T}_j = \mathrm{diag}(\underbrace{\mathfrak{t}^{(i)} - \mathfrak{t}^+, \ldots, \mathfrak{t}^{(i)} - \mathfrak{t}^+}_{j-1 \text{ times}}, \underbrace{\mathfrak{t}^{(i)}, \ldots, \mathfrak{t}^{(i)}}_{d-(j-1) \text{ times}}).$$

It then returns the signature $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. Note that although the signature tag is $\mathfrak{t}^{(i)}$, the effective tag in the preimage sampling is $\mathbf{T}_j$. Since $\mathfrak{t}^+$ is different from all the $\mathfrak{t}^{(i)}$, and since $\mathfrak{t}^{(i)} - \mathfrak{t}^+$ has infinity norm bounded by 2, we can use [LS18] to argue that $\mathfrak{t}^{(i)} - \mathfrak{t}^{+}$ is in $R_q^{\times}$ as desired. Hence, $\mathbf{T}_j \in GL_d(R_q)$. Also, notice that when $j = 1$, we can directly see that $G_{1,0}$ is exactly the game G_1 from before.

Game $G_{j,1}$. This game is the same as $G_{j,0}$ except in the way $\mathbf{A}_3$ is generated. Instead of sampling $\mathbf{A}_3$ uniformly, we hide the gadget $\mathbf{G}_j$ by first sampling $\mathbf{A}_3'$ from $U(R_q^{d \times k})$ and defining $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3 \bmod qR$. In $G_{j,1}$, $\mathbf{A}_3'$ is sampled uniformly and independently of $\mathbf{G}_j$. Thence, $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3' \bmod qR$ is also uniformly distributed, as in $G_{j,0}$. So the views are identically distributed.

Game $G_{j,2}$. We now hide a short relation in $\mathbf{A}_3'$. That is, we sample $\mathbf{R}_j'$ from $\mathcal{B}_1^{2d \times k}$ such that $\mathbf{R}_{-j} = [\mathbf{R}_1 | \ldots | \mathbf{R}_{j-1} | \mathbf{R}_j' | \mathbf{R}_{j+1} | \ldots | \mathbf{R}_d]$ satisfies $\|\mathbf{R}_{-j}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$. It then defines $\mathbf{A}_3' = \mathbf{A}\mathbf{R}_j' \bmod qR$. At this point, the matrix $\mathbf{A}_3$ is now equal to $\mathbf{G}_j - \mathbf{A}\mathbf{R}_j' \bmod qR$.

We now argue that if one distinguishes $G_{j,2}$ from $G_{j,1}$, then it can solve M-LWE. Let $\mathcal{D}$ be a distinguisher between the views from $G_{j,1}$ and $G_{j,2}$. We construct a distinguisher $\mathcal{D}'$ for M-LWE$_{n,d,d,q,\mathcal{B}_1}^{k}$. Given a multiple-secret M-LWE challenge $(\mathbf{A}', \mathbf{A}_3') \in R_q^{d \times d} \times R_q^{d \times k}$, $\mathcal{D}'$ assumes the role of the challenger in the games and uses $\mathbf{A}', \mathbf{A}_3'$ to perfectly simulate the interaction with $\mathcal{A}$. It then sends the resulting view to $\mathcal{D}$. If $\mathcal{D}$ responded $G_{j,1}$, then $\mathcal{D}'$ respond 0 (uniform), and 1 (LWE) if $\mathcal{D}$ responded $G_{j,2}$. Indeed, if $\mathbf{A}_3'$ is uniform, then the view exactly simulate that of $G_{j,1}$, and if $\mathbf{A}_3' = [\mathbf{I}_d | \mathbf{A}']\mathbf{R}_j'$ for some $\mathbf{R}_j' \sim \mathcal{B}_1^{2d \times k}$, then it correctly simulates $G_{j,2}$. As a result, it holds that

$$\forall \mathcal{D} \text{ PPT distinguisher}, \mathsf{Adv}_{G_{j,1},G_{j,2}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}},$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound for M-LWE$_{n,d,d,q,\mathcal{B}_1}$ defined as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{D}'' \text{ PPT}} \mathsf{Adv}_{\text{M-LWE}}[\mathcal{D}'']$. Note that here, we implicitly use a standard hybrid argument showing that M-LWE$_{n,d,m,q,\mathscr{D}_r}^{k}$ is at least as hard as M-LWE$_{n,d,m,q,\mathscr{D}_r}^{1}$ at the expense of a loss factor k in the reduction.

Game $G_{j,3}$. In game $G_{j,3}$, we modify the way signing queries are answered by switching the partial trapdoor $\mathbf{R}_j$ for $\mathbf{R}_j'$. Concretely, upon reception of a message $\mathbf{m}^{(i)} \in T_1^m$, the signer gets the tag $\mathfrak{t}^{(i)}$, sample $\mathbf{v}_{2,j}^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then compute

$$\begin{aligned}&(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, (\mathbf{v}_{2,1}^{(i)}, \ldots, \mathbf{v}_{2,j-1}^{(i)}, \mathbf{v}_3^{(i)}, \mathbf{v}_{2,j+1}^{(i)}, \ldots, \mathbf{v}_{2,d}^{(i)}))\\ &= \mathsf{SamplePre}(\mathbf{R}_{-j}, \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - (\mathfrak{t}^{(i)}\mathbf{G}_j - \mathbf{B}_j)\mathbf{v}_{2,j}^{(i)}, \mathbf{T}_{-j}, s_1, s_2, s_{\mathbf{G}}),\end{aligned}$$

where

$$\mathbf{T}_{-j} = \operatorname{diag}(\underbrace{\mathfrak{t}^{(i)} - \mathfrak{t}^{+}, \ldots, \mathfrak{t}^{(i)} - \mathfrak{t}^{+}}_{j-1 \text{ times}}, 1, \underbrace{\mathfrak{t}^{(i)}, \ldots, \mathfrak{t}^{(i)}}_{d-j \text{ times}}).$$

It then sends the signature $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. Using the trapdoor switching result from Lemma 4.1 on a single query gives a relative error between $\mathcal{P}_1$ and $\mathcal{P}_2$ of $\delta - 1$. Indeed $\mathcal{P}_1/\mathcal{P}_2 - 1 \in [\delta^{-1} - 1, \delta - 1] \subseteq [-(\delta - 1), (\delta - 1)]$. We then use the relative error lemma of Lemma 2.1 and the multiplicativity of the Rényi divergence (of order 2λ) to get

$$\mathsf{Adv}_{G_{j,2}}[\mathcal{A}] \lesssim (1 + Q(\lambda - 1/2)(\delta - 1)^2)\mathsf{Adv}_{G_{j,3}}[\mathcal{A}]^{1-1/2\lambda}.$$

In our scheme, we set δ so that $Q(\lambda - 1/2)(\delta - 1)^2 \leq 1/4$, thus resulting in a factor of less than 5/4. We note that we have $\delta - 1 \sim K\varepsilon$ for $K = 2(12d(n-1)+7) = \mathsf{poly}(\lambda)$. When setting parameters (e.g. $n = 256, d = 4, Q = 2^{32}$), choosing $\varepsilon \approx 2^{-36}$ is sufficient to guarantee $Q(\lambda - 1/2)(\delta - 1)^2 \leq 1/4$ meaning it only incurs a loss of less than half a bit.

Game $G_{j,4}$. By noticing that the partial trapdoor $\mathbf{R}_j$ is no longer used in $G_{j,3}$, we can now simulate the public key $\mathbf{B}_j$. More precisely, we sample $\mathbf{B}_j$ directly from $U(R_q^{d\times k})$. Using the same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{B}_j)$ this time, we obtain

$$\forall \mathcal{D} \text{ } \mathsf{PPT} \text{ distinguisher}, \mathsf{Adv}_{G_{j,3},G_{j,4}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,5}$. We now hide the guess on the forgery tag within the public key $\mathbf{B}_j$. For that, we sample $\mathbf{B}'_j \hookleftarrow U(R_q^{d\times k})$ and define $\mathbf{B}_j = \mathbf{B}'_j + \mathsf{t}^+\mathbf{G}_j$. Since $\mathbf{B}'_j$ is uniform and independent of $\mathsf{t}^+\mathbf{G}_j$, then $\mathbf{B}_j = \mathbf{B}'_j + \mathsf{t}^+\mathbf{G}_j$ is also uniform, as in $G_{j,4}$. So the views are identically distributed.

Game $G_{j,6}$. We then re-hide a short trapdoor in the matrix $\mathbf{B}'_j$. We thus sample $\mathbf{R}_j$ from $\mathcal{B}_1^{2d\times k}$ conditioned on $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$, and then define $\mathbf{B}'_j = \mathbf{A}\mathbf{R}_j \bmod qR$. At this point, the matrix $\mathbf{B}_j$ is now equal to $\mathbf{A}\mathbf{R}_j + \mathsf{t}^+\mathbf{G}_j \bmod qR$. The same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{B}'_j)$ yields

$$\forall \mathcal{D} \text{ } \mathsf{PPT} \text{ distinguisher}, \mathsf{Adv}_{G_{j,5},G_{j,6}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,7}$. In game $G_{j,7}$, we again modify the way signing queries are answered to use the partial trapdoor $\mathbf{R}_j$ instead of $\mathbf{R}'_j$. This means that when receiving $\mathbf{m}^{(i)} \in T_1^m$, the signer gets the tag $\mathsf{t}^{(i)}$, sample $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then compute

$$(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) = \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathbf{T}_{j+1}, s_1, s_2),$$

where

$$\mathbf{T}_{j+1} = \mathrm{diag}(\underbrace{\mathsf{t}^{(i)} - \mathsf{t}^+, \ldots, \mathsf{t}^{(i)} - \mathsf{t}^+}_{j \text{ times}}, \underbrace{\mathsf{t}^{(i)}, \ldots, \mathsf{t}^{(i)}}_{d-j \text{ times}}).$$

and sends the signature $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. As for $G_{j,2}$-$G_{j,3}$, the trapdoor switching lemma the relative error lemma yield

$$\mathsf{Adv}_{G_{j,6}}[\mathcal{A}] \lesssim (1 + Q(\lambda - 1/2)(\delta - 1)^2)\mathsf{Adv}_{G_{j,7}}[\mathcal{A}]^{1-1/2\lambda}.$$

Game $G_{j,8}$. We then remove the short relation in $\mathbf{A}'_3$. That is instead of sampling $\mathbf{R}'_j$ and defining $\mathbf{A}'_3 = \mathbf{A}\mathbf{R}'_j$, we simply sample $\mathbf{A}'_3 \hookleftarrow U(R_q^{d\times k})$. The same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{A}'_3)$ yields

$$\forall \mathcal{D} \text{ } \mathsf{PPT} \text{ distinguisher}, \mathsf{Adv}_{G_{j,7},G_{j,8}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,9}$. We finally remove the gadget from in $\mathbf{A}_3$. Instead of sampling $\mathbf{A}'_3$ uniformly and defining $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}'_3$, we directly sample $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$. Since in $G_{j,8}$, $\mathbf{A}'_3$ is uniform and independent of $\mathbf{G}_j$, then $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}'_3$ is also uniform, as in $G_{j,9}$. So the views are identically distributed.

We can clearly see that $G_{j,9} = G_{j+1,0}$ for $j \in [d-1]$, meaning we can indeed chain these games in a hybrid argument. Additionally, hoping from $G_{j,0}$ to $G_{j,9}$ results in a loss characterized by the following inequality.

$$\mathsf{Adv}_{G_{j,0}}[\mathcal{A}] \lesssim k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(2k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(k\varepsilon_{\text{M-LWE}} + \mathsf{Adv}_{G_{j,9}}[\mathcal{A}]\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}},$$

that is $\mathsf{Adv}_{G_{j,0}}[\mathcal{A}] \lesssim h(\mathsf{Adv}_{G_{j,9}}[\mathcal{A}])$, where

$$h(x) = k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(2k\varepsilon_{\text{M-LWE}} + \frac{5}{4}\left(k\varepsilon_{\text{M-LWE}} + x\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}}.$$

Because h is non-decreasing, looping over all $j \in [d]$ thus gives

$$\mathsf{Adv}_{G_{1,0}}[\mathcal{A}] \lesssim h^{\circ d}(\mathsf{Adv}_{G_{d,9}}[\mathcal{A}]). \tag{2}$$

Although the powers $\frac{2\lambda-1}{2\lambda}$ will stack up with composing the function h d times due to the hybrid argument, the exponent is sufficiently close to 1 and d is a very small integer (typically $d = 4$) so that it only incurs a loss of a few bits, typically around d bits. We give more details on how to bound $h^{\circ d}$ in Appendix A. We thus end up with the following game.

```
G_{d,9}
  Setup
    1. $\forall i \in [Q], \mathsf{t}^{(i)} = F(\mathsf{st} + i - 1)$
    2. $\mathsf{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathsf{t}^{(i)}; i \in [Q]\})$
    3. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d\times dk}$ s.t. $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
    4. $\mathbf{B} \leftarrow \mathbf{AR} + \mathsf{t}^+\mathbf{G} \bmod qR$
    5. $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$
  Queries
    Given $\mathbf{m}^{(i)} \in T_1^m$, get $\mathsf{t}^{(i)}$. Then
    1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$
    2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_3^{(i)}) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)} - \mathsf{t}^+, s_1, s_2)$
    3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.
```

Bounding the advantage. We now need to bound $\mathsf{Adv}_{G_{d,9}}[\mathcal{A}]$. For that we use an adversary in $G_{d,9}$ can be used to construct an adversary $\mathcal{B}$ to solve M-SIS$_{n,d,2d+k+m+1,q,\beta_{❶}}$. Upon reception of the M-SIS instance, $\mathcal{B}$ parses it into $[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3|\mathbf{D}|\mathbf{u}]$ and uses these elements to simulate the challenger in $G_{d,9}$. After the queries stage, it receives a type ❶ forgery from $\mathcal{A}$, i.e., it receives a forgery $\mathsf{sig}^\star = (\mathsf{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ on $\mathbf{m}^\star$ such that $\mathsf{Verify}(\mathsf{pk}, \mathsf{sig}^\star, \mathbf{m}^\star) = 1$. At this point, if $\mathsf{t}^\star \neq \mathsf{t}^+$ then $\mathcal{B}$ aborts which happens with probability $1 - 1/(|\mathcal{T}_w| - Q)$. Then, it also aborts if $\|\mathbf{R}\mathbf{v}_2\|_2 > \frac{1}{\sqrt{2}}\sqrt{2nd}\|\mathbf{v}_2^\star\|_2$. By Lemma 2.4, this happens with probability at most $1 - 1/C$ for a small constant C (typically $C = 2$ in our parameter setting), because $\mathbf{R}$ is hidden in $\mathbf{B}$ under M-LWE. If it did not abort, it computes

$$\mathbf{v}_{1,1}^\star = \mathbf{u} + \mathbf{Dm}^\star - (\mathbf{A}'\mathbf{v}_{1,2}^\star + (\mathsf{t}^\star\mathbf{G} - \mathbf{B})\mathbf{v}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star) \bmod qR,$$

and defines $\mathbf{v}_1^\star = [{\mathbf{v}_{1,1}^\star}^T | {\mathbf{v}_{1,2}^\star}^T]^T$. Since $\mathsf{t}^\star = \mathsf{t}^+$, we have $\mathsf{t}^\star \mathbf{G} - \mathbf{B} = [\mathbf{I}_d | \mathbf{A}'] \mathbf{R} \bmod qR$. Also, as verification passes, we know that $\|\mathbf{v}_1^\star\|_2$, $\|\mathbf{v}_3^\star\|_2$ are bounded by B_1, B_3 respectively. We can re-write the definition of $\mathbf{v}_{1,1}^\star$ as

$$[\mathbf{I}_d | \mathbf{A}' | \mathbf{A}_3 | \mathbf{D} | \mathbf{u}] \mathbf{x}^\star = \mathbf{0} \bmod qR, \text{ where } \mathbf{x}^\star = \begin{bmatrix} \mathbf{v}_1^\star - \mathbf{R}\mathbf{v}_2^\star \\ \mathbf{v}_3^\star \\ \mathbf{m}^\star \\ -1 \end{bmatrix}.$$

It directly holds that $\mathbf{x}^\star \neq \mathbf{0}$ and we have

$$\|\mathbf{x}^\star\|_2^2 \leq \left(B_1 + \sqrt{nd} B_2 \right)^2 + B_3^2 + nm + 1 = \beta_{❶}^2.$$

It thus means that $\mathbf{x}^\star$ is a solution to $\text{M-SIS}_{n,d,2d+k+m+1,q,\beta_{❶}}$ and the advantage of $\mathcal{B}$ is $\mathsf{Adv}_{G_{d,9}}[\mathcal{A}] \cdot (C(|\mathcal{T}_w| - Q))^{-1}$. It in turn gives

$$\mathsf{Adv}_{G_{d,9}}[\mathcal{A}] \leq C(|\mathcal{T}_w| - Q) \varepsilon_{\text{M-SIS}}, \tag{3}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound of M-SIS. Combining Equations (2) and (3) and the fact that h is non-decreasing and that $\mathsf{Adv}_{❶}[\mathcal{A}] = \mathsf{Adv}_{G_{1,0}}[\mathcal{A}]$ yields the result. □

Theorem 5.2. *An adversary produces a forgery* $(\mathsf{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ *of* Type ❷ *if the tag* $\mathsf{t}^\star$ *is re-used from some* $i^\star$*-th signing query* $(\mathsf{t}^{(i^\star)}, \mathbf{v}_{1,2}^{(i^\star)}, \mathbf{v}_2^{(i^\star)}, \mathbf{v}_3^{(i^\star)})$*. The advantage of any* PPT *adversary* $\mathcal{A}$ *in producing a type* ❷ *forgery is at most*

$$Adv_{❷}[\mathcal{A}] \lesssim m\varepsilon_{\text{M-LWE}} + 2MC \frac{1+\varepsilon}{1-\varepsilon} h^{\circ d}(QC^2 \varepsilon_{\text{M-SIS}}) + \mathsf{negl}(\lambda).$$

where C *is a small constant from Lemma 2.4,* $\varepsilon_{\text{M-SIS}}$ *is the hardness bound of* $\text{M-SIS}_{n,d,d(2+k),q,\beta_{❷}}$ *for*

$$\beta_{❷} = \sqrt{(2B_1 + 2\sqrt{nd}B_2 + n\sqrt{dm})^2 + 4B_2^2}.$$

and h *is the function of Theorem 5.1 depending on* $\varepsilon_{\text{M-LWE}}$ *which is the hardness bound of* $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$.

Proof. Throughout the proof, we consider an PPT adversary $\mathcal{A}$ interacting with the challenger $\mathcal{B}$, and which aims at producing a valid Type ❷ forgery. We proceed by a game hop to modify the distribution of the view of $\mathcal{A}$ in a way that is indistinguishable from the real distribution. In the last game, the constructed elements that compose the distribution given to $\mathcal{A}$ allow to easily exploit the forgery to obtain a solution to M-SIS. Under the assumption that M-SIS is hard, it should thus be infeasible for $\mathcal{A}$ to produce a valid type ❷ forgery. We again proceed using a game-based proof which follows the sequence summarized in Figure 5.2.

Games Hops. We define the following games which are composed of three stages: setup, queries, forgery. Past the queries stage, the view of the adversary

```
– $G_0$                                              ▷ Original game
– $G_1$                                              ▷ Sample tags at the start
– $G_2$                                              ▷ Hiding short relation in $\mathbf{D}$
– $G_3$                                              ▷ Simulating $\mathbf{u}$
– $G_4$                                              ▷ Enforcing norm bounds
– $G_5$                                              ▷ Adding rejection
– $G_6$                                              ▷ Simulating $i^+$-th query
– For $j \in [d]$
–     For $i \in [0, 9]$                             ▷ Hiding tag guess in partial key $j$
–         $G_{j,i}$
– Solve M-SIS using $\mathcal{A}$ against $G_{d,9}$.
```

Fig. 5.2. Overview of the unforgeability reduction (type ❷)

does not change so we only describe the first two stages. The matrix $\mathbf{A}'$ (and in turn $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}'])$ is always generated the same way, i.e., $\mathbf{A}' \hookleftarrow U(R_q^{d \times d})$, and we thus do not specify them in the games below. In each game, the view of $\mathcal{A}$ is $(\mathbf{A}, \mathbf{D}, \mathbf{B}, \mathbf{u}, \mathbf{A}_3, (\mathsf{sig}^{(i)})_{i \in [Q]})$.

Game G_0. This corresponds to the original unforgeability game where the key material generation and signing queries are handled honestly. More precisely, we have

```
$G_0$
  Setup
    1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d \times dk}$ s.t. $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
    2. $\mathbf{B} = \mathbf{A}\mathbf{R} \bmod qR$
    3. $\mathbf{A}_3 \hookleftarrow U(R_q^{d \times k})$
    4. $\mathbf{D} \hookleftarrow U(R_q^{d \times m})$
    5. $\mathbf{u} \hookleftarrow U(R_q^{d})$
  Queries
    Given $\mathbf{m}^{(i)} \in T_1^m$, compute $\mathsf{t}^{(i)} = F(\mathsf{st})$ and increment $\mathsf{st}$. Then
    1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k, s_2}$
    2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)}, s_1, s_2, s_{\mathbf{G}})$
    3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.
```

Game G_1. In G_1, we simply change the way tags are generated. Instead of computing $\mathsf{t}^{(i)}$ at each signing query, we first generate and store all the Q tags during the setup stage. In the query stage, we simply look-up the corresponding tag. In addition, we make a guess on the tag that will be used in the forgery (although it is not used at this point). More precisely, we sample $i^+ \hookleftarrow U([Q])$ and define $\mathsf{t}^+ = \mathsf{t}^{(i^+)}$. The view is exactly the same in G_1 because we only changed the moment when the tags are generated. Since they are generated deterministically from the state, and since the tag guess t^+ does not intervene, both views are identically distributed.

Game G_2. We now hide a short relation in $\mathbf{D}$. More precisely, we sample $\mathbf{S}$ from $\mathcal{B}_1^{2d \times m}$ and define $\mathbf{D} = \mathbf{A}\mathbf{S} \bmod qR$. We now argue that if one distinguishes G_2 from G_1, then it can solve M-LWE. Let $\mathcal{D}$ be a distinguisher between the views from G_1 and G_2. We construct a distinguisher $\mathcal{D}'$ for $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}^m$. Given a multiple-secret M-LWE challenge $(\mathbf{A}', \mathbf{D}) \in R_q^{d \times d} \times R_q^{d \times m}$, $\mathcal{D}'$ assumes the role of the challenger in the games and uses $\mathbf{A}', \mathbf{D}$ to perfectly simulate the interaction with $\mathcal{A}$. It then sends the resulting view to $\mathcal{D}$. If $\mathcal{D}$ responded G_1, then $\mathcal{D}'$ respond 0 (uniform), and 1 (LWE) if $\mathcal{D}$ responded G_2. Indeed, if $\mathbf{D}$ is uniform, then the view exactly simulate that of G_1, and if $\mathbf{D} = [\mathbf{I}_d|\mathbf{A}']\mathbf{S}$ for some $\mathbf{S} \sim \mathcal{B}_1^{2d\times m}$, then it perfectly simulates G_2. As a result, it holds that

$$\forall \mathcal{D} \text{ } \mathsf{PPT} \text{ distinguisher}, \mathsf{Adv}_{G_1,G_2}[\mathcal{D}] \leq m\varepsilon_{\text{M-LWE}},$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound for $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$ defined as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{D}'' \text{ } \mathsf{PPT}} \mathsf{Adv}_{\text{M-LWE}}[\mathcal{D}'']$.

Game G_3. We then change the way $\mathbf{u}$ is generated by hiding a short relation within it. Concretely, we sample $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{2d},s_1}$, $\mathbf{v}_2, \mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^{dk},s_2}$, and $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^{dk},s_2}$ and define

$$\mathbf{u} = \mathbf{A}\mathbf{v}_1 + (\mathfrak{t}^+\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 \bmod qR.$$

To argue that it is well distributed, we use the regularity lemma from Lemma 2.6. We indeed define $\overline{\mathbf{A}} = [\mathbf{A}|\mathfrak{t}^+\mathbf{G} - \mathbf{B}|\mathbf{A}_3]$ and $\mathbf{v} = [\mathbf{v}_1^T|\mathbf{v}_2^T|\mathbf{v}_3^T]^T$. The covariance matrix of $\mathbf{v}$ is $\text{diag}(s_1^2\mathbf{I}_{2nd}, s_2^2\mathbf{I}_{2nkd})$. By our conditions on s_1, s_2 obtained for the correctness of preimage sampling, we have $s_1 > s_2 \geq \eta_\varepsilon(\mathcal{L}_q^\perp(\overline{\mathbf{A}}))$, where ε is the same used to set $r = \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})$. For the range given by Lemma 2.6, we can obtain the inverse and thus get

$$\mathsf{Adv}_{G_2}[\mathcal{A}] \in [1/(1+\varepsilon), (1+\varepsilon)/(1-\varepsilon)]\mathsf{Adv}_{G_3}[\mathcal{A}].$$

Game G_4. In this step, we enforce a bound on the i^+-th query and aborting if this bound is not verified. Concretely, for $i = i^+$, when receiving $\mathbf{m}^{(i^+)}$ the reduction aborts if $\left\|\mathbf{S}\mathbf{m}^{(i^+)}\right\|_2 > \sqrt{nd}\left\|\mathbf{m}^{(i^+)}\right\|_2$. If it did not abort, it handles the rest of the query as before. As $\mathbf{S}$ is hidden within $\mathbf{D}$ under M-LWE, Lemma 2.4 yields that the norm constraint is verified with a probability negligibly close to $1/C$ for a small constant C (typically $C = 2$ in our parameter setting). We thus get

$$\mathsf{Adv}_{G_4}[\mathcal{A}] = \left(\frac{1}{C} - \mathsf{negl}(\lambda)\right)\mathsf{Adv}_{G_3}[\mathcal{A}].$$

Game G_5. Now, we add the main rejection in the i^+-th query only to anticipate the next game. For $i \neq i^+$, the queries are handled honestly, while for $i = i^+$ we proceed as follows after the norm check introduced in G_4. The signer samples $\mathbf{v}_3^{(i^+)} \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then computes

$$(\mathbf{v}_{1,1}^{(i^+)}, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}) = \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i^+)} - \mathbf{A}_3\mathbf{v}_3^{(i^+)}, \mathfrak{t}^+, s_1, s_2, s_{\mathbf{G}}),$$

which so far is as usual. Then, it samples a continuous $\rho \hookleftarrow U([0,1))$. Now, the reduction continues only if $\rho \leq 1/M$ and if $\langle \mathbf{v}_1 \text{ }, \text{ } \mathbf{S}\mathbf{m}^{(i^+)}\rangle \geq 0$. We insist on the fact that at this point $\mathbf{v}_1$ is the one used to define $\mathbf{u}$ which is different from $\mathbf{v}_1^{(i^+)}$.

First, since ρ is independent from the rest, the first condition is verified with probability $1/M$. Then, since the distribution of $\mathbf{S}$ is centered and because $\mathbf{v}_1$

is hidden in $\mathbf{u}$, the probability that $\langle \mathbf{v}_1 \,,\; \mathbf{S}\mathbf{m}^{(i^+)} \rangle$ is non-negative is negligibly close to $1/2$ as $\mathcal{A}$ cannot predict the sign of $\mathbf{v}_1$ from $\mathbf{u}$. All in all, it means that

$$\mathsf{Adv}_{G_5}[\mathcal{A}] = \left(\frac{1}{2M} - \mathsf{negl}(\lambda) \right) \mathsf{Adv}_{G_4}[\mathcal{A}].$$

Game G_6. We now change how the i^+-th signing query is answered. Upon receiving $\mathbf{m}^{(i^+)}$, the challenger samples $\rho \hookleftarrow U([0,1))$ and computes $\delta = \langle \mathbf{v}_1 + \mathbf{S}\mathbf{m}^{(i^+)} \,,\; \mathbf{S}\mathbf{m}^{(i^+)} \rangle$. Then, it aborts the reduction if

$$\delta < 0 \text{ or } \rho > \frac{1}{M} \exp\left(\frac{\pi}{s_1^2} \left(\left\| \mathbf{S}\mathbf{m}^{(i^+)} \right\|_2^2 - 2\delta \right) \right).$$

If it did not abort, it sets

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

and sends the signature $\mathsf{sig}^{(i^+)} = (\mathbf{t}^+, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}, \mathbf{v}_3^{(i^+)})$.

We now use the rejection sampling result of Lemma 2.9 to argue on the views of G_5 and G_6. For that we simply need to ensure that $s_1 \geq \alpha \left\| \mathbf{S}\mathbf{m}^{(i^+)} \right\|_2$ for $M = \exp(\pi/\alpha^2)$. This is subsumed by the condition

$$s_1 \geq \alpha \cdot \sqrt{nd} \cdot \sqrt{nm},$$

as we enforce the bound on $\mathbf{S}\mathbf{m}^{(i^+)}$. For the correctness and security of sampling, we also need $s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$. Depending on the value of m, we choose s_1 and α as follows. If $\sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6) > \sqrt{\pi/\ln(2)} \cdot n\sqrt{dm}$, we set $s_1 = \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$, and

$$\alpha = \frac{s_1}{n\sqrt{dm}} = \frac{\sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)}{n\sqrt{dm}}.$$

On the other hand, if the inequality is not verified we set $\alpha = \sqrt{\pi/\ln(2)}$, and

$$s_1 = \alpha n \sqrt{dm},$$

which indeed satisfies the sampler's requirements as we have

$$s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6).$$

In both cases, this ensures that $s_1 \geq \alpha \left\| \mathbf{S}\mathbf{m}^{(i^+)} \right\|_2$ for some $\alpha \geq \sqrt{\pi/\ln(2)}$. Note however that in the first case, it can lead to α much larger than $\sqrt{\pi/\ln(2)}$ if m

is small, which in turn yields a smaller repetition rate M. Both conditions can be expressed as

$$\begin{aligned} s_1 &= \max\left(\sqrt{\frac{\pi}{\ln(2)}}n\sqrt{dm}, \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)} \cdot \frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)\right), \\ \alpha &= \frac{s_1}{n\sqrt{dm}}. \end{aligned}$$

Based on these parameter constraints, we use Lemma 2.9 to argue that conditioned on not aborting, the distributions are identical. Hence, the view of $\mathcal{A}$ in G_5 and G_6 are identical.

From the previous game hops, we already have

$$\mathsf{Adv}_{❷}[\mathcal{A}] \leq m\varepsilon_{\text{M-LWE}} + 2MC\frac{1+\varepsilon}{1-\varepsilon}\mathsf{Adv}_{G_6}[\mathcal{A}] + \mathsf{negl}(\lambda). \tag{4}$$

At this point, we use the same hybrid argument that of the proof of Theorem 5.1. That is we are aiming to replace the public key $\mathbf{B} = \mathbf{AR}$ by $\mathbf{B} = \mathbf{AR} + \mathsf{t}^+\mathbf{G}$. In order to do so while keeping the ability answer signing queries for $i \neq i^+$, we use the exact same sequence of games $G_{j,0}$ to $G_{j,9}$ for $j \in [d]$ but by keeping the modifications we made up to G_6. Since the trapdoor is not used in the i^+-th query, we are able to perform these modifications. To avoid repetition we only briefly explain the sequence, recalling that $\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{S}, \mathbf{D}$ and the i^+-th query are unchanged in all of them. Game $G_{j,0}$ corresponds to G_6 but where the public key is $\mathbf{B} = [\mathbf{AR}_1 + \mathsf{t}^+\mathbf{G}_1|\ldots|\mathbf{AR}_{j-1} + \mathsf{t}^+\mathbf{G}_{j-1}|\mathbf{AR}_j|\ldots|\mathbf{AR}_d]$. The signing queries for $i \neq i^+$ are answered with the tag matrix

$$\mathbf{T}_j = \operatorname{diag}(\underbrace{\mathsf{t}^{(i)} - \mathsf{t}^+, \ldots, \mathsf{t}^{(i)} - \mathsf{t}^+}_{j-1 \text{ times}}, \underbrace{\mathsf{t}^{(i)}, \ldots, \mathsf{t}^{(i)}}_{d-(j-1) \text{ times}}).$$

Then in $G_{j,1}$ we introduce a gadget in $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3'$. In $G_{j,2}$, we introduce a short relation as $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{AR}_j'$ under M-LWE. In $G_{j,3}$, we change the partial trapdoor $\mathbf{R}_j$ for $\mathbf{R}_j'$ under the trapdoor switching lemma. In $G_{j,4}$ we simulate the public key $\mathbf{B}_j$ by sampling it uniformly under M-LWE. We then add the tag guess in $G_{j,5}$ as $\mathbf{B}_j = \mathbf{B}_j' + \mathsf{t}^+\mathbf{G}_j$. In $G_{j,6}$ we re-introduce the short relation as $\mathbf{B}_j = \mathbf{AR}_j + \mathsf{t}^+\mathbf{G}_j$ under M-LWE. Then, in $G_{j,7}$ we use the trapdoor switching once more to use $\mathbf{R}_j$ instead of $\mathbf{R}_j'$. In $G_{j,8}$, we simulate $\mathbf{A}_3'$ and get $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3'$ under M-LWE. Finally, in $G_{j,9}$ we remove the gadget and simply sample $\mathbf{A}_3$ uniformly.

Using the exact same reasoning, we have $G_{1,0} = G_6$, $G_{j,9} = G_{j+1,0}$ for all $j \in [d-1]$, and it holds that $\mathsf{Adv}_{G_{j,0}}[\mathcal{A}] \lesssim h(\mathsf{Adv}_{G_{j,9}}[\mathcal{A}])$ where h is the same function as that of Theorem 5.1. As a result, we get

$$\mathsf{Adv}_{G_6}[\mathcal{A}] \lesssim h^{\circ d}(\mathsf{Adv}_{G_{j,9}}[\mathcal{A}]). \tag{5}$$

We end up with the following game.

```
$G_{d,9}$
Setup
  1. $\forall i \in [Q], \mathsf{t}^{(i)} = F(\mathsf{st} + i - 1)$
  2. $i^+ \hookleftarrow U([Q]), \mathsf{t}^+ = \mathsf{t}^{(i^+)}$
  3. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d \times dk}$ s.t. $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
  4. $\mathbf{B} \leftarrow \mathbf{AR} + \mathsf{t}^+\mathbf{G} \bmod qR$
  5. $\mathbf{A}_3 \hookleftarrow U(R_q^{d \times k})$
  6. $\mathbf{S} \hookleftarrow \mathcal{B}_1^{2d \times m}$
  7. $\mathbf{D} \leftarrow \mathbf{AS} \bmod qR$
  8. $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{2d}, s_1}, \mathbf{v}_2 \hookleftarrow \mathcal{D}_{R^{dk}, s_2}, \mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^{dk}, s_2}$
  9. $\mathbf{u} \leftarrow \mathbf{A}\mathbf{v}_1 + (\mathsf{t}^+\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 \bmod qR$
Queries
  Given $\mathbf{m}^{(i)} \in T_1^m$, get $\mathsf{t}^{(i)}$. Then
  If $i \neq i^+$:
    1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k, s_2}$
    2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_3^{(i)}) \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)} - \mathsf{t}^+, s_1, s_2)$
    3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.
  If $i = i^+$:
    1. If $\left\|\mathbf{S}\mathbf{m}^{(i^+)}\right\|_2 > \sqrt{nd}\left\|\mathbf{m}^{(i^+)}\right\|_2$, then abort
    2. $\rho \hookleftarrow U((0,1))$
    3. $\delta \leftarrow \langle \mathbf{v}_1 + \mathbf{S}\mathbf{m}^{(i^+)} \,,\; \mathbf{S}\mathbf{m}^{(i^+)} \rangle$
    4. If $\delta < 0$ or if $\rho > \frac{1}{M} \exp\left(\frac{\pi}{s_1^2}\left(\left\|\mathbf{S}\mathbf{m}^{(i^+)}\right\|_2^2 - 2\delta\right)\right)$, then abort.
    5. Otherwise, set $\begin{bmatrix} \mathbf{v}_{1,1}^{(i+)} \\ \mathbf{v}_{1,2}^{(i^+)} \end{bmatrix} = \mathbf{v}_1 + \mathbf{S}\mathbf{m}^{(i^+)}$, and $\mathbf{v}_2^{(i^+)} = \mathbf{v}_2$, $\mathbf{v}_3^{(i^+)} = \mathbf{v}_3$.
    6. Send $\mathsf{sig}^{(i^+)} = (\mathsf{t}^+, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}, \mathbf{v}_3^{(i^+)})$.
```

Bounding the advantage. We now need to bound $\mathsf{Adv}_{G_{d,9}}[\mathcal{A}]$. For that we use an adversary in $G_{d,9}$ can be used to construct an adversary $\mathcal{B}$ to solve M-SIS$_{n,d,2d+k,q,\beta_{❷}}$. Given the M-SIS instance, $\mathcal{B}$ parses it into $[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3]$ and uses these elements to simulate the challenger in $G_{d,9}$. After the queries stage, it receives a type ❷ forgery from $\mathcal{A}$, i.e., it receives a forgery $\mathsf{sig}^\star = (\mathsf{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ on $\mathbf{m}^\star$ such that $\mathsf{Verify}(\mathsf{pk}, \mathsf{sig}^\star, \mathbf{m}^\star) = 1$. At this point, if $\mathsf{t}^\star \neq \mathsf{t}^+$ then $\mathcal{B}$ aborts which happens with probability $1 - 1/Q$. Then, it also aborts if $\|\mathbf{R} \cdot \Delta\mathbf{v}_2\|_2 > \sqrt{nd}\|\Delta\mathbf{v}_2\|_2$ or $\|\mathbf{S} \cdot \Delta\mathbf{m}\|_2 > \sqrt{nd}\|\Delta\mathbf{m}\|_2$, where $\Delta\mathbf{v}_2 = \mathbf{v}_2^{(i^+)} - \mathbf{v}_2^\star$ and $\Delta\mathbf{m} = \mathbf{m}^{(i^+)} - \mathbf{m}^\star$. Because $\mathbf{R}, \mathbf{S}$ are independent and hidden in $\mathbf{B}$ and $\mathbf{D}$ respectively under M-LWE, Lemma 2.4 gives that the bounds are verified with probability at least $1/C^2$ for a small constant C (typically $C = 2$ in our parameter setting). Hence this step aborts with probability at most $1 - 1/C^2$. If it did not abort, it computes

$$\mathbf{v}_{1,1}^\star = \mathbf{u} + \mathbf{D}\mathbf{m}^\star - (\mathbf{A}'\mathbf{v}_{1,2}^\star + (\mathsf{t}^\star\mathbf{G} - \mathbf{B})\mathbf{v}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star) \bmod qR,$$

and defines $\mathbf{v}_1^\star = [\mathbf{v}_{1,1}^{\star\,T}|\mathbf{v}_{1,2}^{\star\,T}]^T$. Since $\mathsf{t}^\star = \mathsf{t}^+$, we have that $\mathsf{t}^\star\mathbf{G} - \mathbf{B} = -[\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$. Since verification passes, we know that $\|\mathbf{v}_1^\star\|_2, \|\mathbf{v}_3^\star\|_2$ are bounded by B_1, B_3 respectively. Then, by definition of $\mathbf{u}$ and the i^+-th query seen by the attacker, we can re-write this equation as

$$\mathbf{A}\mathbf{v}_1^\star - \mathbf{AR}\mathbf{v}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star = \mathbf{A}(\mathbf{v}_1^{(i^+)} - \mathbf{S}\mathbf{m}^{(i^+)}) - \mathbf{AR}\mathbf{v}_2^{(i^+)} + \mathbf{A}_3\mathbf{v}_3^{(i^+)} + \mathbf{AS}\mathbf{m}^\star \bmod qR,$$

which leads to

$$\begin{aligned}&[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3]\mathbf{x}^\star = \mathbf{0} \bmod qR,\\ &\text{where } \mathbf{x}^\star = \begin{bmatrix}(\mathbf{v}_1^{(i^+)} - \mathbf{v}_1^\star) - \mathbf{R}(\mathbf{v}_2^{(i^+)} - \mathbf{v}_2^\star) - \mathbf{S}(\mathbf{m}^{(i^+)} - \mathbf{m}^\star)\\ \mathbf{v}_3^{(i^+)} - \mathbf{v}_3^\star\end{bmatrix}.\end{aligned}$$

There, we use the same argument as in [JRS23] to argue that $\mathbf{x}^\star \neq \mathbf{0}$ with overwhelming probability. More precisely, since $\mathbf{m}^{(i^+)} \neq \mathbf{m}^\star$, at least one column $\mathbf{s}^\star$ of $\mathbf{S}$ appears in $\mathbf{x}^\star$. Yet, $\mathbf{S}$ is hidden in $\mathbf{D}$ at the exception of at most one bit due to the rejection sampling leak of the sign of δ. As is done in [LNS21,LNP22], under an extended version of M-LWE which is proven to be at least as hard as M-LWE, $\mathbf{s}^\star$ is unpredictable resulting in $\mathbf{x}^\star \neq \mathbf{0}$ with overwhelming probability. Finally, it holds that

$$\|\mathbf{x}^\star\|_2^2 \leq \left(2B_1 + \sqrt{nd}\cdot 2B_2 + \sqrt{nd}\cdot\sqrt{nm}\right)^2 + (2B_3)^2 = \beta_{❷}^2,$$

where the inequality holds based on the Gaussian tail bound from Lemma 2.7 and the Johnson-Lindenstrauss bound from Lemma 2.4 we enforced.

It thus means that $\mathbf{x}^\star$ is a solution to M-SIS$_{n,d,2d+k,q,\beta_{❷}}$ and the advantage of $\mathcal{B}$ is at least $\mathsf{Adv}_{G_{d,9}}[\mathcal{A}]/(QC^2)$. It in turn gives

$$\mathsf{Adv}_{G_{d,9}}[\mathcal{A}] \leq QC^2\varepsilon_{\text{M-SIS}} + \mathsf{negl}(\lambda), \tag{6}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound. Combining Equations (4), (5), (6), and the fact that h is non-decreasing, yields the result. □

6 Anonymous Credentials

The protocols associated to the signature scheme, namely the issuance for obtaining a signature on a hidden message and the showing for proving knowledge of a valid credential, follow the same blueprint as [JRS23]. We recall them in Algorithms 6.3 and 6.4. We note that we could define generic protocols $\mathsf{OblSign}$ and Prove as in [JRS23] to encompass a larger variety of applications, but we decide to focus on the anonymous credentials. Adapting them to other contexts is fairly straightforward. As opposed to the original paper, the signer does not need to contribute to the commitment randomness during issuance for the security proof to hold. So we only have to introduce the randomness from the user to ensure the hiding property of the commitment. As is done in [JRS23], we can use the matrix $\mathbf{A}$ as the commitment matrix in order to merge the randomness with the vector $\mathbf{v}_1$. However, we do not need to use a Gaussian randomness. The authors of [JRS23] chose a Gaussian distribution in order to have a statistically hiding commitment. Here, to improve the efficiency, we aim at a computationally hiding commitment and choose $\mathbf{r}$ to be uniform over T_1^{2d}. Under the M-LWE assumption, $\mathbf{Ar}$ is indeed indistinguishable from uniform. This means that the signature the user obtains features $\mathbf{v}_1 - \mathbf{r}$. As a result, we need to adjust the verification bound on $\mathbf{v}_1$. In addition, we also need to slightly adjust the rejection sampling condition on s_1 for the reduction to go through because the randomness from the user is now part of the vector we perform rejection sampling on in the i^+-th query. As such we change Algorithm 5.1 with

$$s_1 = \max\left(\sqrt{\frac{\pi}{\ln 2}}\left(n\sqrt{dm}+\sqrt{2nd}\right), \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2-1}}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)\right),$$
$$\alpha = \frac{s_1}{n\sqrt{dm}+\sqrt{2nd}},$$

and the verification bound becomes $B_1' = B_1 + \sqrt{2nd} = c_{2nd}s_1\sqrt{2nd}+\sqrt{2nd}$. To avoid confusion, we call $\mathsf{Verify}'(\mathsf{pk}, \mathbf{m}, (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3); \mathsf{pp})$ the modified verification where the bound B_1 is replaced by B_1'.

6.1 The Construction

The resulting anonymous credentials is then very similar in design to that of [JRS23]. The differences between our schemes is mainly due to the underlying signature, and a more precise description of the zero-knowledge proof system.

Algorithm 6.1: OKeyGen

```
Input: Public parameters pp as in Algorithm 5.1.
Output: $(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$.        ▷ Algorithm 5.2
```

Algorithm 6.2: UKeyGen

```
Input: Public parameters pp as in Algorithm 5.1.
1. $\mathbf{s} \hookleftarrow U(T_1^{m_s})$.                              ▷ $m_s = 2d$
2. $\mathbf{t} \leftarrow \mathbf{D}_s\mathbf{s} \bmod qR$.
Output: $(\mathsf{upk}, \mathsf{usk}) = (\mathbf{t}, \mathbf{s})$.
```

Algorithm 6.3: Issue (Credential Issuance Protocol)

```
Input: Organization $O$ with $\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st}$, and a user $U$ with $\mathbf{m} \in T_1^m$
and $\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}$.
    User $U$.
1.  $\mathbf{r} \hookleftarrow U(T_1^{2d})$.
2.  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{r} + \mathbf{D}_s\mathbf{s} + \mathbf{D}\mathbf{m} \bmod qR$.
3.  Send $\mathbf{c}$ to $O$.
    User $U \longleftrightarrow$ Organization $O$.
4.  Interactive zero-knowledge argument between $U$ and $O$. In this syntax,
    i.e., [FHS19], the organization knows $\mathbf{m}$ but not usk. Hence, in the ZKAoK,
    $U$ proves knowledge of short $(\mathbf{r}, \mathbf{s})$ such that $\mathbf{c} - \mathbf{D}\mathbf{m} = \mathbf{A}\mathbf{r} + \mathbf{D}_s\mathbf{s} \bmod qR$,
    and additionally that $\mathbf{D}_s\mathbf{s} = \mathsf{upk} \bmod qR$. If $O$ is not convinced, the protocol
    aborts. The zero-knowledge argument is described in Section 7.1.
    Organization $O$.
5.  $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
6.  $\mathsf{t} \leftarrow F(\mathsf{st})$.
7.  $\mathbf{v}' \leftarrow \mathsf{SamplePre}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{c} - \mathbf{A}_3\mathbf{v}_3, \mathsf{t}, s_1, s_2)$.
```

```
8. Parse $\mathbf{v}' = [{\mathbf{v}'_{1,1}}^T | {\mathbf{v}'_{1,2}}^T | \mathbf{v}_2^T]^T$
9. Send $(\mathsf{t}, \mathbf{v}'_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ to $U$.
10. $\mathsf{st} \leftarrow \mathsf{st} + 1$
    User $U$.
11. Parse $\mathbf{r}$ as $[\mathbf{r}_{1,1}^T | \mathbf{r}_{1,2}^T]^T$ with $\mathbf{r}_{1,i} \in R^d$.
12. $\mathbf{v}_{1,2} \leftarrow \mathbf{v}'_{1,2} - \mathbf{r}_{1,2}$.
13. if $\mathsf{Verify}'(\mathsf{pk}; \mathbf{m}; (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3); \mathsf{pp}) = 1$, then return $(\tau, \mathbf{v})$.    ▷ Algorithm 5.4
14. else return $\perp$
```

Algorithm 6.4: Show (Credential Showing Protocol)

```
Input: User $U$ with $\mathsf{usk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}, \mathsf{sig}, \mathcal{I}$, 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 $(\mathbf{s}, (\mathbf{m}_i)_{i \notin \mathcal{I}}; \mathsf{sig})$ such that $\mathsf{Verify}'(\mathsf{pk}, \widetilde{\mathbf{m}}, \mathsf{sig}, \mathsf{pp}) = 1$. The zero-knowledge argument is described in Section 7.2.
```

6.2 Security Analysis

We now provide the security proofs of the anonymous credentials for completeness, even though it follows the same proof structure as that of [JRS23]. Notice that similarly to [JRS23], and as opposed to the constructions of [BLNS23] and [LLLW23], we do not require straightline extractable proofs. This is because the security proof of the anonymous credentials presented in Algorithms 6.1 to 6.4 only requires to extract one issuance proof corresponding to the tag guess, and one show proof to extract the forgery. The proof techniques from these other constructions [BLNS23,LLLW23] require straightline extraction as they essentially need to extract every issuance proof to detect a forgery.

We give the formal statements for the correctness, anonymity and unforgeability in Lemma 6.1, 6.2, and 6.3.

Lemma 6.1. *The anonymous credentials system of Algorithms 6.1 to 6.4 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 $\mathbf{m} \in T_1^m$ and $\mathcal{I} \subseteq [m]$. We consider an honest execution of the issuance protocol $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$. From the completeness of the zero-knowledge argument of knowledge, we only have to check the abort condition of step 13. First, note that $\mathsf{t} \in \mathcal{T}_w$ and $\widetilde{\mathbf{m}} \in T_1^{m+m_s}$. Then, we define $\mathbf{v}_{1,1} = \mathbf{u} + \mathbf{D}\mathbf{m} + \mathbf{D}_s\mathbf{s} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR$. As the signature was honestly generated, it holds that $\mathbf{v}_1 = \mathbf{v}'_1 - \mathbf{r}$ and that $(\mathbf{v}'_1, \mathbf{v}_2)$ were obtained by a call to $\mathsf{SamplePre}$. Lemma 3.3 ensures that the distribution of $(\mathbf{v}'_1, \mathbf{v}_2)$ is a factor within $[\delta_1, \delta_2]$ of the elliptical discrete Gaussian (we call it $\mathcal{D}$ here for clarity) over the appropriate coset. We thus have that

$$\begin{aligned}\mathbb{P}_{\mathbf{v}'_1,\mathbf{v}_2\leftarrow\mathsf{SamplePre}}&[\|\mathbf{v}'_1\|_2 > B_1 \vee \|\mathbf{v}_2\|_2 > B_2]\\ &\leq \delta_2 \cdot \mathbb{P}_{\mathbf{v}'_1,\mathbf{v}_2\leftarrow\mathcal{D}}[\|\mathbf{v}'_1\|_2 > B_1 \vee \|\mathbf{v}_2\|_2 > B_2]\\ &\leq \delta_2(2^{-(\lambda+O(1))} + 2^{-(\lambda+O(1))})\\ &= \delta_2 \cdot 2^{-(\lambda+O(1))},\end{aligned}$$

where the second inequality follows from the Gaussian tail bound of Lemma 2.7 and the union bound. As $\mathbf{v}_3$ is sampled directly from $\mathcal{D}_{R^k,s_2}$, we can also apply Lemma 2.7 and get that it is bounded by B_3 except with probability at most $2^{-(\lambda+O(1))}$. Combining it all, along with the triangle inequality $\|\mathbf{v}_1\|_2 \leq \|\mathbf{v}'_1\|_2 + \|\mathbf{r}\|_2 \leq \|\mathbf{v}'_1\|_2 + \sqrt{2nd}$, we obtain

$$\mathbb{P}_{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3}[\|\mathbf{v}_1\|_2 > B'_1 \vee \|\mathbf{v}_2\|_2 > B_2 \vee \|\mathbf{v}_3\|_2 > B_3] \leq (\delta_2+1)2^{-(\lambda+O(1))} = \mathsf{negl}(\lambda),$$

as desired.

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 it did not abort, it means that the outputted credential passed verification, i.e., that $\mathsf{Verify}'(\mathsf{pk}, \widetilde{\mathbf{m}}, \mathsf{sig}, \mathsf{pp}) = 1$. The completeness of the zero-knowledge argument then yields 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 $(\perp, 1)$. □

Lemma 6.2. *The anonymous credentials of Algorithms 6.1 to 6.4 is anonymous based on the zero-knowledge property of the proof system of Section 7.2.*

Proof. The proof follows the same idea as that of [JRS23]. It simply consists in simulating the zero-knowledge proof in the Show interaction in the anonymity game, relying on the zero-knowledge property of the proof system. More formally, we define the modified game to be exactly that of Figure 2.2 except that when interacting with $\mathcal{A}$ 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 $\mathcal{C}$ simulates the zero-knowledge argument, i.e., without resorting to usk_{j_b}, $(\mathbf{m}_i^{(j'_b)})_{i\notin\mathcal{I}}$, $\mathsf{cred}^{(j'_b)}$. By Lemma 7.6, the advantage of $\mathcal{A}$ in the modified game is negligibly close to that it would have been in the original game under the assumption that $\text{M-LWE}_{\widehat{n},m_2-(\widehat{d}+256/\widehat{n}+\ell+1),m_2,\widehat{q},\mathcal{B}_1}$ is hard.

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. □

Lemma 6.3. *The anonymous credentials of Algorithms 6.1 to 6.4 is unforgeable based on the hardness of* $\text{M-ISIS}_{d,m_s,q,\sqrt{nm_s}}$*, the zero-knowledge and soundness properties of the proof systems of Section 7.1 and 7.2, and on the EUF-CMA security of the signature scheme of Section 5.*

Proof. The proof follows the exact same blueprint and distinguish two types of forgeries: (1) impersonation forgeries, and (2) credential forgeries (either tampering with the proof or by forging a signature). The first case relies on the Lemma 7.3, 7.6 and Lemma 7.5 and the M-ISIS assumption on the matrix $\mathbf{D}_s$. The second relies on the Lemma 7.2, 7.5 and the EUF-CMA security of the signature captured by Theorems 5.1 and 5.2. For the sake of completeness, we provide the proof by following the proof structure of [JRS23].

We consider a PPT adversary $\mathcal{A}$ against the unforgeability game. It receives opk and gives a set of disclosed attributes $\mathbf{m}^*_{\mathcal{I}}$ while proving possession of a credential cred^* on said attributes in a successful execution of Show with the honest organization. If $\mathbf{m}^*_{\mathcal{I}}$ corresponds to an attribute vector $\mathbf{m}$ that was queried for issuance by a corrupt user, the forgery is not valid. We thus have two possible cases: (1) $\mathcal{A}$ tried to impersonate an honest user, or (2) it did not. As $\mathcal{A}$ must convince a the challenger they know a secret $\mathbf{s}^*$ satisfying $\mathbf{D}_s\mathbf{s}^* = \mathbf{t}$, this means that (1) corresponds to the scenario 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.

(1) Impersonation Forgery. The challenger receives the M-ISIS instance $(\overline{\mathbf{D}_s}, \overline{\mathbf{t}})$. It then runs Setup by setting $\mathbf{D}_s = \overline{\mathbf{D}_s}$ instead of sampling it themselves. It then makes a guess on which honest user will be targeted. For that it samples $j^+ \hookleftarrow U([|\mathcal{T}_w|])$. As in [JRS23], the number of users requesting credentials to the organization is bounded by the number of possible tags, which is polynomial. It then runs $\mathsf{OKeyGen}(\mathsf{pp})$ to obtain $(\mathsf{opk}, \mathsf{osk}) = ((\mathbf{B}, \mathbf{u}), \mathbf{R})$, and sends opk to $\mathcal{A}$. We now describe how the oracle queries are answered.

- $\mathcal{O}_{\mathsf{HU}}$: Given an 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 $j = j^+$.
- $\mathcal{O}_{\mathsf{CU}}$: Given 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.
- $\mathcal{O}_{\mathsf{ObtIss}}$: Given j and an attribute vector $\mathbf{m} \in T_1^m$, it sends $\perp$ to $\mathcal{A}$ if $j \in \mathsf{CU}$. 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. If $j = j^+$, it instead generates $\mathbf{c}$ as $\mathbf{Ar} + \overline{\mathbf{t}} + \sum_i \mathbf{D}_i\mathbf{m}_i \bmod qR$, and simulates the zero-knowledge argument when assuming the role of the user in Step 4 of Issue. By Lemma 7.3, this is unnoticeable by the adversary. Again, if this modified execution fails, it sends $\perp$ to $\mathcal{A}$, and nothing if it succeeds.
- $\mathcal{O}_{\mathsf{Issue}}$: Given j and an attribute vector $\mathbf{m} \in T_1^m$, it returns $\perp$ to $\mathcal{A}$ and does not engage in the issuance protocol if $j \notin \mathsf{CU}$. Otherwise, since the challenger knows osk, it can run the Issue protocol where the adversary embodies the user j with public key upk_j, and the challenger embodies the signer. Then, either $\mathcal{A}$ gets $\perp$ if the execution failed, or obtained a credential sig on $\mathbf{m}$.
- $\mathcal{O}_{\mathsf{Show}}$: Given an issuance index j' corresponding to the j'-th credential issued on $\mathbf{m}^{(j')}$ for some user j, and also disclosed attributes $\mathbf{m}^{(j')}_{\mathcal{I}}$, the challenger outputs $\perp$ to $\mathcal{A}$ if $j \in \mathsf{CU}$. 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. By Lemma 7.6, this remains unnoticeable by $\mathcal{A}$.

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 $\bar{\mathbf{t}}$ is uniform, it is indistinguishable from regular keys $\mathbf{D}_s\mathbf{s}$ under M-LWE$_{d,d,q,U(T_1)}$. Hence, if $\mathcal{A}$ has advantage δ in performing a forgery attack satisfying (1), it can successfully prove knowledge of $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}^*}, \mathsf{sig}^*)$ with disclosed attributes $\mathbf{m}_{\mathcal{I}}^*$ such that $\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}^*, \mathsf{sig}^*, \mathsf{pp}) = 1$ where $\widetilde{\mathbf{m}}^* = [\mathbf{s}^{*T}|\mathbf{m}^{*T}]^T$. The challenger then extracts $\mathbf{s}^*$ by Lemma 7.5. As it verifies the conditions of (1), there must exist $j^* \in \mathsf{HU}$ 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 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 $\overline{\mathbf{D}_s}\mathbf{s}^* = \bar{\mathbf{t}} \bmod qR$, and $\mathbf{s}^* \in T_1^{m_s}$ yielding $\|\mathbf{s}^*\|_2 \le \sqrt{nm_s}$. The challenger thus solves the M-ISIS instance with advantage at least $\delta/|\mathcal{T}_w| - \mathsf{negl}(\lambda)$.

(2) Credential Forgery. If the challenger expects this type of forgery, it expects a forgery on the signature scheme of Section 5. It therefore tosses a coin to guess which of type ❶ or type ❷ the forgery will be. Note that the M-SIS bounds underlying the security against those forgeries are updated to use B_1' instead of B_1.

If it expects a type ❶ forgery, it proceeds exactly as in the proof of Theorem 5.1, 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 $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}}, \mathsf{sig}^*)$ with disclosed attributes $\mathbf{m}_{\mathcal{I}}^*$ such that $\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}^*, \mathsf{sig}^*, \mathsf{pp}) = 1$, the challenger can extract $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}}, \mathsf{sig}^*)$ by Lemma 7.5. Then, sig^* is a valid type ❶ forgery for the signature as $\widetilde{\mathbf{m}}^*$ is a fresh message. Indeed, by definition of type (2) forgeries, 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 anonymous credentials, it must hold that for all $j \in \mathsf{CU}$, $(j, j', \mathbf{m}^*) \notin \mathsf{A})$, 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 can invoke Theorem 5.1, thus relying on M-LWE and M-SIS.

If it expects a type ❷ forgery of the signature, it proceeds as in the proof of Theorem 5.2 with the only difference that it needs to control the commitment randomness for the i^+-th signature query. In this context, in the issuance corresponding to the tag $\mathbf{t}^* = \mathbf{t}^{(i^+)}$ that will be used in the forgery extracted from the showing, the challenger proceeds as follows. By Lemma 7.2, it extracts $(\mathbf{r}^{(i^+)}, \mathbf{s}^{(i^+)})$ such that $\mathbf{c}^{(i^+)} = \mathbf{A}\mathbf{r}^{(i^+)} + \mathbf{D}_s\mathbf{s}^{(i^+)} + \mathbf{D}\mathbf{m}^{(i^+)} \bmod qR$. As opposed to the proof of Theorem 5.2 where it performed rejection on $\mathbf{v}_1^{(i^+)} = \mathbf{v}_1 + \mathbf{S}\widetilde{\mathbf{m}}^{(i^+)}$, with $\widetilde{\mathbf{m}}^{(i^+)} = [{\mathbf{s}^{(i^+)}}^T | {\mathbf{m}^{(i^+)}}^T]^T$, here, it performs rejection on

$$\mathbf{v}_1^{(i^+)} = \mathbf{v}_1 + \mathbf{S}\widetilde{\mathbf{m}}^{(i^+)} + \mathbf{r}^{(i^+)}.$$

The rest of the proof remains the same. In the end, when $\mathcal{A}$ engages in Show to attack the unforgeability of the anonymous credentials, the challenger extracts $(\mathbf{s}^*, (\mathbf{m}_i^*)_{i \notin \mathcal{I}}, \mathsf{sig}^*)$. It thus obtain a valid type ❷ forgery for the SEP on message $\widetilde{\mathbf{m}}^*$ (which is fresh as explained above). We can thus invoke Theorem 5.2, thus relying on M-LWE and 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 $\delta \leq \sup_{\mathcal{A}' \text{ PPT}} \mathrm{Adv}^{\text{EUF-CMA}}[\mathcal{A}'] + \mathsf{negl}(\lambda)$, where $\mathrm{Adv}^{\text{EUF-CMA}}[\mathcal{A}']$ denotes the advantage of $\mathcal{A}'$ in producing a valid forgery (type ❶ or type ❷) for our signature. □

6.3 Extensions

Although it is not explicitly done in our construction, we mention here several possible extensions, e.g., to support pseudonyms, relations between attributes, or ways to realize revocation.

Pseudonyms. Our protocol has been designed to match the anonymous credential model from [FHS19] perfectly. We nevertheless note that it could easily be adapted to the one in [LLLW23] where users can unlinkably request anonymous credentials for different pseudonyms. Indeed, the commitment $\mathbf{c}$ and proof of opening of our issuance protocol could play the role of a pseudonym, as is done in [LLLW23], and we could split this protocol into a "registration" one and an "issue" one to match the syntax of [LLLW23]. Unlinkability would readily follow from the hiding property of the commitment.

Relations between Attributes. Many use-cases of anonymous credentials require proving statements about attributes (e.g. age > 18) without revealing them. The features of the zero-knowledge proof system we use [LNP22] allow for doing that at almost no cost as long as these statements can be associated to linear or quadratic relations. Note that this is for example sufficient for range proofs (see, e.g., [CBC+24]) which address the age control use-case. More complex statements may require extending the witness, which would impact the proof size. We share this functionality with [JRS23,LLLW23] as we do not hash the attributes (as opposed to [BCR+23,BLNS23]).

Revocation. Revocation is usually not considered by anonymous credentials systems. More generally, revoking anonymous signatures is notoriously difficult but this can be circumvented by adding appropriate elements in the signature. Typically, DAA [BCC04] includes a tag which deterministically depends on the user's secret key and a so-called basename. This allows for some forms of traceability which can be leveraged to revoke users. Alternatively, EPID systems define a list of revoked signatures. Each new signature must then include a proof by the signer that it did not generate any of the revoked signatures which thus acts as a proof of non-revocation. Given the similarities between these constructions and ours (in particular their reliance on SEP) one could easily add these functionalities to our system by using the same techniques as in [CKLL19,BEF19].

7 Zero-Knowledge Arguments

We now give the zero-knowledge arguments needed in the anonymous credentials. For that, we follow the blueprint of [LNP22] and detail the arguments for (1) the proof of opening and proof of registration (for Algorithm 6.3) and (2) the proof of credential possession (for Algorithm 6.4).

This situation is typical of anonymous credentials (and related primitives) and sometimes leads to extractibility issues where the reduction would have to rewind several zero-knowledge proofs (potentially in parallel) to extract all the witnesses. This is specifically the case for (1) when one wants to prove unforgeability under the EUF-CMA security of the underlying signature scheme: one needs to "decapsulate" the committed messages so as to submit them to the EUF-CMA oracle and this is usually done through extraction of the corresponding witnesses. In such cases, one either needs to bound the number of parallel executions of the protocol (which is only possible in the interactive setting) or resort to straight-line extractable zero-knowledge proofs which are more complex. The latter strategy was chosen in [LLLW23] for example which actually presents it as an advantage over the state-of-the-art.

We however stress that the proof strategy of [JRS23] and therefore of our construction is *not* concerned by those extractibility issues. It indeed does not exactly rely on the EUF-CMA security of the signature scheme but directly on the underlying assumption. Most importantly, it only requires to extract *one* commitment opening proof and is thus immune to the problems stemming from parallel rewindings.

The case of step (2) is harder to consider in general but we note that most models allow to clearly identify the zero-knowledge proof that needs to be extracted. This is exactly the situation in our case: we only need to extract the one zero-knowledge proof that is produced by the adversary when it "proves" authenticity of a set of attributes for which it never received a credential. As a consequence, we do not need straight-line extractable proof as our reduction only needs to perform two rewindings [LNP22].

We nevertheless note that this low number of rewindings is still too much if one targets security in the UC framework [Can20]. In such a case, one would need to switch to straight-line extractable proofs, which would negatively impact performance.

As opposed to the result of [JRS23], we make use of subrings to improve the zero-knowledge proof size. As explained in [LNPS21] and recalled in Section 2.1, using a smaller ring reduces size of elements that are not dependent on the witness dimension, and thus reduces the overall proof size. To benefit from this improvement while keeping compact keys for the signature scheme, we consider a ring R of degree n for the signature, and a subring $\widehat{R}$ of degree $\widehat{n}|n$ for the zero-knowledge proof. We explain for each protocol exactly how to use the subring embedding θ and M_θ of Section 2.1 to map relations over R into relations over $\widehat{R}$.

7.1 Proof of Commitment Opening and User Registration

In Algorithm 6.3, the user needs to prove knowledge of a commitment opening as well as the secret key associated to its public key (which we call user registration). We present the argument so that the attributes remain hidden even though it differs from the presentation of Algorithm 6.3. Revealing the message will only make the proof simpler and smaller, so we deal with the worst case where everything must be concealed.

Relation. The relation entails proving knowledge of $\mathbf{r} \in R^{2d}$, $\mathbf{m} \in R^m$ and $\mathbf{s} \in R^{m_s}$ such that

$$\mathbf{A}\mathbf{r} + \mathbf{D}\mathbf{m} = \mathbf{c} - \mathsf{upk} \bmod qR \text{ and } \mathbf{D}_s\mathbf{s} = \mathsf{upk} \bmod qR$$
$$\mathbf{r} \in T_1^{2d}, \mathbf{s} \in T_1^{m_s}, \mathbf{m} \in T_1^m$$

where $\mathbf{A} \in R_q^{d\times 2d}$, $\mathbf{D}_s \in R_q^{d\times m_s}$, $\mathbf{D} \in R_q^{d\times m}$, $\mathbf{c} \in R_q^d$, and $\mathsf{upk} \in R_q^d$ are public elements part of the statement. To prove such a statement, we first lift the equation to $R_{\widehat{q}}$ where $\widehat{q} = q_1 q$ is the modulus for the proof system. We require that q_1 has the same splitting behavior as q, that is $q_1 = 2\kappa + 1 \bmod 4\kappa$, and we also need q_1 large enough so that difference of challenges are invertible modulo $\widehat{q}$. Then, since all the vectors must be proven binary, we compact everything into a single equation. We also use the subring embedding θ and M_θ of Section 2.1 to map the relation to $\widehat{R}$. Recall that using M_θ, proving the linear relation $\mathbf{M}\mathbf{x} = \mathbf{y} \bmod \widehat{q}R$ is equivalent to proving $M_\theta(\mathbf{M})\theta(\mathbf{x}) = \theta(\mathbf{y}) \bmod \widehat{q}\widehat{R}$. In the end, we prove the following.

$$\mathbf{C}\mathbf{s}_1 = \mathbf{u} \bmod \widehat{q}\widehat{R}, \quad \text{and} \quad \mathbf{s}_1 \in \widehat{T}_1^{m_1},$$

where $\mathbf{s}_1 = [\theta(\mathbf{r})^T|\theta(\mathbf{s})^T|\theta(\mathbf{m})^T]^T$, $m_1 = \widehat{k}(2d + m_s + m)$, and

$$\mathbf{C} = q_1 M_\theta\left(\begin{bmatrix} \mathbf{A} & \mathbf{0}_{d\times m_s} & \mathbf{D} \\ \mathbf{0}_{d\times 2d} & \mathbf{D}_s & \mathbf{0}_{d\times m}\end{bmatrix}\right), \quad \text{and} \quad \mathbf{u} = q_1\theta\left(\begin{bmatrix} \mathbf{c} - \mathsf{upk} \\ \mathsf{upk}\end{bmatrix}\right).$$

Then, we define $q_{\min} = \min(q_1, q)$, and let $\ell = \lceil \lambda / \log_2 q_{\min} \rceil$ the parameter used for soundness amplification, i.e., so that $q_{\min}^{-\ell} \leq 2^{-\lambda}$.

Challenge Space. We use the same family of challenge spaces as [LNP22]. Recall that for any element $c = \sum_{0\leq i<\widehat{n}} c_i x^i$, its conjugate is defined as $c^* =$

$c(x^{-1}) = c_0 - \sum_{i \in [\widehat{n}-1]} c_{\widehat{n}-i} x^i$. For vectors and matrices, the superscript denotes the conjugate transpose. The conjugate operator corresponds to the automorphism σ_{-1} in [LNP22]. One can see that if $c^* = c$ then $c_i = -c_{\widehat{n}-i}$ for all $i \in [\widehat{n} - 1]$, thus implying $c_{\widehat{n}/2} = 0$. We define

$$\mathcal{C}' = \{c \in \widehat{S}_\rho : c^* = c\},$$

where ρ is a positive integer. The challenge space is defined by

$$\mathcal{C} = \{c \in \mathcal{C}' : \sqrt[2k']{\|c^{2k'}\|_1} \leq \eta\},$$

where η is a positive integer, and k' is a power-of-two that we later choose to be $k' = 32$. From the observation above, we have $|\mathcal{C}'| = (2\rho + 1)^{\widehat{n}/2}$. We thus choose ρ so that this size is at least $2^{\lambda+1}$, that is

$$\rho = \left\lceil \frac{1}{2} \left(2^{2(\lambda+1)/\widehat{n}} - 1 \right) \right\rceil.$$

Then, we determine η heuristically so that $\mathbb{P}_{c \sim U(\mathcal{C}')}[\sqrt[2k']{\|c^{2k'}\|_1} \leq \eta] \geq 1/2$. As a result, we would end up with $|\mathcal{C}| \geq 2^\lambda$. The challenge space places the constraint $q_{\min} > (2\rho\sqrt{\kappa})^\kappa$ which is almost always verified for typical parameters as κ is chosen to be either 2 or 4, and ρ is also small.

The Protocol. *First Round.* We start by the main commitment phase which consists in committing to the witness, masks and randomness needed in subsequent rounds. We sample $\mathbf{s}_2$ from χ^{m_2} where $Supp(\chi) \subseteq \widehat{S}_1$ and compute an Ajtai commitment of $\mathbf{s}_1$ with randomness $\mathbf{s}_2$ as $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$, where $\mathbf{A}_1 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d} \times m_1})$ and $\mathbf{A}_2 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d} \times m_2})$ are part of the common reference string crs. Then, we sample the Gaussian masks for what will later be $c\mathbf{s}_1$ and $c\mathbf{s}_2$. More precisely, we sample $\mathbf{y}_1$ from $\mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$ and $\mathbf{y}_2$ from $\mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$, and compute the commitment $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$.

We then sample a mask $\mathbf{y}_3$ from $\mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$ and a vector for soundness amplification by $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$ where all the entries are polynomials with a constant coefficient equal to zero. We later use $\widehat{\mathbf{m}}$ to denote the vector $\widehat{\mathbf{m}} = [\mathbf{y}_3^T | \mathbf{g}^T]^T \in \widehat{R}^{256/\widehat{n}+\ell}$. We commit to it via $\mathbf{t}_B = \mathbf{B}_{y,g}\mathbf{s}_2 + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$, where $\mathbf{B}_{y,g} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{(256/\widehat{n}+\ell) \times m_2})$ is part of crs.

The prover sends msg_1 as the first message and receives chal_1 as the first challenge, where they are both defined as

$$\mathsf{msg}_1 = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}) \in \widehat{R}_{\widehat{q}}^{2\widehat{d}+256/\widehat{n}+\ell}$$
$$\mathsf{chal}_1 = \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) = (\mathsf{R}_0, \mathsf{R}_1) \in \left(\{0,1\}^{256 \times m_1 \widehat{n}}\right)^2$$

Second Round. Now, we conclude the approximate range proof part. We define $\mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$. The challenge R is used to project the witness onto a smaller dimensional space and prove that the coefficients of $\mathbf{s}_1$ are small relative to $\widehat{q}$. In the second round, we respond to the challenge by masking $\mathsf{R}\tau(\mathbf{s}_1)$ with $\tau(\mathbf{y}_3)$. So we compute $\mathbf{z}_3^{\mathbb{Z}} = \tau(\mathbf{y}_3)+\mathsf{R}\tau(\mathbf{s}_1) \in \mathbb{Z}^{256}$. Then, we perform rejection by sampling $u_3 \hookleftarrow U([0,1))$ and rejecting if

$$u_3 > \frac{1}{M_3} \exp\left(\pi \frac{-2\langle \mathbf{z}_3^{\mathbb{Z}}\ ,\ \mathsf{R}\tau(\mathbf{s}_1)\rangle + \|\mathsf{R}\tau(\mathbf{s}_1)\|_2^2}{\sigma_3^2}\right).$$

If the u_3 is smaller, then the prover accepts, sends msg_2 as the second message and receives chal_2 as the second challenge where they are defined by

$$\begin{aligned}\mathsf{msg}_2 &= \mathbf{z}_3^{\mathbb{Z}} \in \mathbb{Z}^{256}\\ \mathsf{chal}_2 &= \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) = (\gamma_{i,j})_{\substack{i\in[\ell]\\ j\in[257]}} \in \mathbb{Z}_{\widehat{q}}^{\ell\times 257}.\end{aligned}$$

<u>*Third Round.*</u> We now need to prove the equations over $\mathbb{Z}_{\widehat{q}}$, namely the ones related to norm constraints or binary vectors, as well as the well-formedness of $\mathbf{z}_3^{\mathbb{Z}}$.

$$\tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) = \mathbf{z}_3^{\mathbb{Z}}, \tag{3.1}$$

$$\langle \tau(\mathbf{s}_1)\ ,\ \tau(\mathbf{s}_1) - \mathbf{1}_{\widehat{n}m_1}\rangle = 0, \tag{3.2}$$

As observed in [LNP22], the equation $\langle \mathbf{x}\ ,\ \mathbf{y}\rangle = 0$ is equivalent to

$$\tau_0((\tau^{-1}(\mathbf{x}))^*\tau^{-1}(\mathbf{y})) = 0$$

which allows us to interpret $\mathbb{Z}_{\widehat{q}}$-equations as $\widehat{R}_{\widehat{q}}$-equations with automorphisms instead. We write $\mathbf{1}_{\widehat{R}^\delta} = \tau^{-1}(\mathbf{1}_{\widehat{n}\delta}) = [\sum_{i=0}^{\widehat{n}-1} X^i]_{j\in[\delta]}$. We also write $\mathbf{e}_j^{\mathbb{Z}}$ to be the j-th canonical vector of $\mathbb{Z}^{\delta\widehat{n}}$, where the dimension δ is implicit, and let $\mathbf{e}_j = \tau^{-1}(\mathbf{e}_j^{\mathbb{Z}}) \in \widehat{R}^\delta$. As a contrast, we later write $\mathbf{e}_j^{\widehat{R}}$ to be the j-th canonical vector of $\widehat{R}^\delta$ for a rank δ implicit, that is $\mathbf{e}_j^{\widehat{R}}$ has a 1 at position j and 0 elsewhere. The equations above are thus equivalent to

$$\forall j \in [256], \tau_0\left(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}\right) = 0, \tag{3.1*}$$

$$\tau_0\left(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})\right) = 0, \tag{3.2*}$$

where $\mathsf{r}_j = \tau^{-1}(\mathsf{R}^T\mathbf{e}_j^{\mathbb{Z}})$. We combine all of these quadratic equations with automorphisms by computing elements h_i for each $i \in [\ell]$ as follows

$$h_i = g_i + \sum_{j\in[256]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})) \tag{7}$$

The prover then sends msg_3 as the third message and receives chal_3 as the third challenge, where they are both defined as

$$\begin{aligned}\mathsf{msg}_3 &= (h_1, \ldots, h_\ell) \in \widehat{R}_{\widehat{q}}^\ell\\ \mathsf{chal}_3 &= \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) = (\mu_i)_{i\in[\ell+2d\widehat{k}]} \in \widehat{R}_{\widehat{q}}^{\ell+2d\widehat{k}}.\end{aligned}$$

Fourth Round. All the quadratic equations, including the equations for the well-formedness of the h_i for the quadratic evaluations from the previous round, are then proven in round 4 directly over $\widehat{R}_{\widehat{q}}$. We need to prove that the h_i are well-formed and equal their expressions above, and we also need to prove the linear relation $\mathbf{C}\mathbf{s}_1 = \mathbf{u}$. The latter represent $2d\widehat{k}$ equations. We prove them all at once by combining them linearly with the challenges μ_i and prove that

$$\begin{aligned}0 = &\sum_{i \in [\ell]} \mu_i \left(g_i + \sum_{j \in [256]} \gamma_{i,j}(\mathbf{e}_j^* \mathbf{y}_3 + \mathsf{r}_j^* \mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})) - h_i \right) \\ &+ \sum_{i \in [2d\widehat{k}]} \mu_{\ell+i} \left({\mathbf{e}_i^{\widehat{R}}}^T \mathbf{C}\mathbf{s}_1 - u_i \right).\end{aligned}$$

For that let us define $\widehat{\mathbf{s}} = [\mathbf{s}_1^T | \mathbf{s}_1^* | \widehat{\mathbf{m}}^T | \widehat{\mathbf{m}}^*]^T$. Then, the equation to be proven is equivalent to $\widehat{\mathbf{s}}^T \mathbf{F} \widehat{\mathbf{s}} + \mathbf{f}^T \widehat{\mathbf{s}} + f = 0 \bmod \widehat{q}\widehat{R}$, where

$$\begin{aligned} f &= -\sum_{i \in [\ell]} \mu_i \left(\sum_{j \in [256]} \gamma_{i,j} z_{3,j}^{\mathbb{Z}} + h_i \right) - \sum_{i \in [2d\widehat{k}]} \mu_{\ell+i} u_i \\ \mathbf{f} &= \begin{bmatrix} \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_j^{*T} + \sum_{i \in [2d\widehat{k}]} \mu_{\ell+i} \mathbf{C}^T \mathbf{e}_i^{\widehat{R}} \\ -\sum_{i \in [\ell]} \mu_i \gamma_{i,257} \mathbf{1}_{\widehat{R}^{m_1}} \\ \sum_{i \in [\ell]} \mu_i \sum_{j \in [256]} \gamma_{i,j} \mathbf{e}_j^{*T} \\ [\mu_1 | \ldots | \mu_\ell]^T \\ \mathbf{0}_{256/\widehat{n}} \\ \mathbf{0}_\ell \end{bmatrix} \qquad (8) \\ \mathbf{F} &= \begin{bmatrix} \mathbf{0}_{m_1 \times m_1} \ \sum_{i \in [\ell]} \mu_i \gamma_{i,257} \mathbf{I}_{m_1} \ \mathbf{0}_{m_1 \times 2(256/\widehat{n}+\ell)} \\ \mathbf{0}_{m_1 + 2(256/\widehat{n}+\ell) \times 2(m_1 + 256/\widehat{n}+\ell)} \end{bmatrix}, \end{aligned}$$

Once we have defined these (public) elements, we can compute the garbage terms and commit to them. More precisely, we define

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_1^{*T} \\ -\mathbf{B}_{y,g}\mathbf{y}_2 \\ -(\mathbf{B}_{y,g}\mathbf{y}_2)^{*T} \end{bmatrix} \in \widehat{R}_{\widehat{q}}^{2(m_1+256/\widehat{n}+\ell)}, \qquad (9)$$

and compute $e_0 = \mathbf{y}^T \mathbf{F} \mathbf{y} \bmod \widehat{q}\widehat{R}$, $e_1 = \widehat{\mathbf{s}}^T \mathbf{F} \mathbf{y} + \mathbf{y}^T \mathbf{F} \widehat{\mathbf{s}} + \mathbf{f}^T \mathbf{y}$, and the commitments $t_0 = \mathbf{b}^T \mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$ and $t_1 = \mathbf{b}^T \mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$, where $\mathbf{b} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{m_2})$ is part of crs. The prover then sends msg_4 as the fourth message and receives chal_4 as the fourth challenge, where they are both defined as

$$\begin{aligned} \mathsf{msg}_4 &= (t_0, t_1) \in \widehat{R}_{\widehat{q}}^2 \\ \mathsf{chal}_4 &= \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c \in \mathcal{C}. \end{aligned}$$

Fifth Round. The final round consists in the final response that is typical of Schnorr-like proofs, i.e., outputting $z = y + c \cdot s$ without it leaking information.

More precisely, the prover responds to the challenge by masking $c\mathbf{s}_1$ and $c\mathbf{s}_2$ with $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively. So we compute $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$. Then, we perform rejection by sampling $u_1, u_2 \hookleftarrow U([0,1))$ and rejecting if

$$\begin{aligned} u_1 &> \frac{1}{M_1} \exp\left(\pi \frac{-2\langle \tau(\mathbf{z}_1)\ ,\ \tau(c\mathbf{s}_1)\rangle + \|\tau(c\mathbf{s}_1)\|_2^2}{\sigma_1^2} \right) \\ \text{or } u_2 &> \frac{1}{M_2} \exp\left(\pi \frac{-2\langle \tau(\mathbf{z}_2)\ ,\ \tau(c\mathbf{s}_2)\rangle + \|\tau(c\mathbf{s}_2)\|_2^2}{\sigma_2^2} \right). \end{aligned}$$

If the u_1, u_2 are both smaller than these respective bounds, then the prover accepts, sends msg_5 as the final message defined by

$$\mathsf{msg}_5 = (\mathbf{z}_1, \mathbf{z}_2) \in \widehat{R}^{m_1+m_2}.$$

<u>*Verification.*</u> Upon receiving msg_5, the verifier computes $\mathbf{F}, \mathbf{f}, f$, as well as

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_1^{*T} \\ c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2 \\ (c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2)^{*T} \end{bmatrix}, \tag{10}$$

and then checks the following six conditions.

$$\|\mathbf{z}_1\|_2 \leq c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}, \|\mathbf{z}_2\|_2 \leq c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2}, \|\mathbf{z}_3^{\mathbb{Z}}\|_2 \leq c_{256}\sigma_3\sqrt{256} \tag{11}$$

$$\forall i \in [\ell], \tau_0(h_i) = 0 \tag{12}$$

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = \mathbf{w} + c\mathbf{t}_A \bmod \widehat{q}\widehat{R} \tag{13}$$

$$\mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f - (ct_1 - \mathbf{b}^T\mathbf{z}_2) = t_0 \bmod \widehat{q}\widehat{R}. \tag{14}$$

Transcript and Communication Complexity. The transcript is thus composed of the five messages and four challenges. Note that in the interactive setting, the challenges are selected uniformly in their respective space and not computed from $\mathcal{H}$. The hash function $\mathcal{H}$ is presented here if one desires to make the proof non-interactive.

The total size of the messages send by the prover to the verifier can be details as follows. The elements $\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}, h_1, \ldots, h_\ell, t_0, t_1$ cannot be compressed as they all[15] look uniformly random modulo $\widehat{q}$. To evaluate the size of the discrete Gaussian vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3^{\mathbb{Z}}$, we use the entropy bound which can be achieved using the rANS encoding as discussed [ETWY22]. More precisely, for a discrete Gaussian over $\mathbb{Z}^N$ of width s, the estimated bit size is $N(1/2 + \log_2 s)$. It means the total bit-size of the message part can be estimated by

$$\begin{aligned} &\left(2\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 2\right)\widehat{n}\lceil \log_2 \widehat{q} \rceil + \widehat{n}m_1(1/2 + \log_2 \sigma_1) \\ &+ \widehat{n}m_2(1/2 + \log_2 \sigma_2) + 256(1/2 + \log_2 \sigma_3). \end{aligned}$$

[15] The elements $h_1, \ldots, h_\ell$ have a constant coefficient equal to zero, so it may not be necessary to send this coefficient.

For the challenges, the maximal bit-size can be easily bounded by

$$2 \cdot 256 \cdot m_1 \widehat{n} + (\ell(256+3) + (2d\widehat{k} + \ell)\widehat{n})\lceil \log_2 \widehat{q} \rceil + \widehat{n}\lceil \log_2(2\rho+1) \rceil.$$

As $\mathbf{w}$, t_0 and the challenges can be re-computed from the rest, the proof can be condensed to $\pi = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$ in the non-interactive case. In that case, the overall proof size can be bounded by

$$\begin{aligned}|\pi| \leq &\left(\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 1\right)\widehat{n}\lceil \log_2 \widehat{q} \rceil + \widehat{n}m_1(1/2 + \log_2 \sigma_1) \\ &+ \widehat{n}m_2(1/2 + \log_2 \sigma_2) + 256(1/2 + \log_2 \sigma_3) + \widehat{n}\lceil \log_2(2\rho+1) \rceil.\end{aligned}$$

Security Analysis. The proof of completeness from Lemma 7.1 follows by the rejection sampling result of Lemma 2.8, the tail bound of Lemma 2.7 and careful inspection of the verification equations with respect to the committed variables. We detail the proof for reference.

Lemma 7.1. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i) \cdot (\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \alpha_1 \eta \sqrt{\widehat{n}m_1}$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n}m_2}$ and $\sigma_3 = \alpha_3 \sqrt{337}\sqrt{\widehat{n}m_1}$. Then, the (interactive) zero-knowledge argument in Figure 7.1 is complete.*

Proof. First we look at the correctness of the different rejection sampling step and the probability of aborting. First, we have that $\|c\mathbf{s}_1\|_2 \leq \sqrt[2k']{\|c^{2k'}\|_1} \cdot \|\mathbf{s}_1\|_2 \leq \eta\sqrt{\widehat{n}m_1}$ as $\mathbf{s}_1 \in \widehat{T}_1^{m_1}$. By setting α_1, σ_1 as in the lemma statement, Lemma 2.8 yields that this step aborts with a probability that is within $[1 - 1/M_1, 1 - 1/M_1 + \varepsilon_1/M_1]$ and that the real distribution is within statistical distance ε_1/M_1 and Rényi divergence $1 + \varepsilon_1/(M_1 - 1)$. Similarly, we have $\|c\mathbf{s}_2\|_2 \leq \eta\sqrt{\widehat{n}m_2}$. The same reasoning gives that this step aborts with probability that is within $[1 - 1/M_2, 1 - 1/M_2 + \varepsilon_2/M_2]$ and that the real distribution is within statistical distance ε_2/M_2 and Rényi divergence $1+\varepsilon_2/(M_2-1)$. Then, by [LNP22, Lem. 2.8] we have $\|\mathsf{R}\tau(\mathbf{s}_1)\|_2 \leq \sqrt{337}\sqrt{\widehat{n}m_1}$ with overwhelming probability (heuristically). So the rejection sampling step on $\mathbf{z}_3^{\mathbb{Z}}$ with probability that is within $[1-1/M_3, 1-1/M_3+\varepsilon_3/M_3]$ and that the real distribution is within statistical distance ε_3/M_3 and Rényi divergence $1+\varepsilon_3/(M_3-1)$. The probability of not aborting is therefore negligibly close to $1/(M_1M_2M_3)$.

Now we study the completeness of non-aborting transcripts. It holds by the previous arguments that

$$\begin{aligned}&\mathbb{P}_{\mathbf{z}_1}[\|\mathbf{z}_1\|_2 > c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}] \\ &\quad\leq \mathbb{P}_{\mathbf{z}_1 \sim \mathcal{D}_{\widehat{R}^{m_1}, \sigma_1}}[\|\mathbf{z}_1\|_2 > c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}] \cdot \left(1 + \frac{\varepsilon_1}{M_1 - 1}\right) \\ &\quad\leq 2^{-(\lambda + O(1))} \cdot \left(1 + \frac{\varepsilon_1}{M_1 - 1}\right) \\ &\quad= \mathsf{negl}(\lambda),\end{aligned}$$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
$\mathbf{s}_2 \hookleftarrow \chi^{m_2}$ $\mathbf{t}_A \leftarrow \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$ $\mathbf{y}_i \hookleftarrow \mathcal{D}_{\widehat{R}^{m_i},\sigma_i}$ for $i \in \{1,2\}$ $\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$ $\mathbf{y}_3 \hookleftarrow \mathcal{D}_{\widehat{R}^{256/n},\sigma_3}$ $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^{\ell})$ $\mathbf{t}_B \leftarrow \mathbf{B}_{y,g}\mathbf{s}_2 + [\mathbf{y}_3^T \vert \mathbf{g}^T]^T \bmod \widehat{q}\widehat{R}$		
	$\xrightarrow{(\mathbf{t}_A, \mathbf{t}_B, \mathbf{w})}$	
		$(\mathsf{R}_0, \mathsf{R}_1) \hookleftarrow U(\{0,1\}^{256 \times m_1 \widehat{n}})^2$
	$\xleftarrow{(\mathsf{R}_0, \mathsf{R}_1)}$	
$\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$ $\mathbf{z}_3^{\mathbb{Z}} \leftarrow \tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1)$ $\perp$ **if** $\mathsf{Rej}_1(\mathbf{z}_3^{\mathbb{Z}}, \mathsf{R}\tau(\mathbf{s}_1), \sigma_3, M_3) = 0$		
	$\xrightarrow{\mathbf{z}_3^{\mathbb{Z}}}$	
		$(\gamma_{i,j})_{i,j} \hookleftarrow U(\mathbb{Z}_{\widehat{q}}^{\ell \times 257})$
	$\xleftarrow{(\gamma_{i,j})_{i,j}}$	
Compute all h_i as in Equation (7)		
	$\xrightarrow{(h_1, \ldots, h_\ell)}$	
		$(\mu_i)_{i \in [\ell + 2d\widehat{k}]} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\ell + 2d\widehat{k}})$
	$\xleftarrow{(\mu_i)_{i \in [\ell + 2d\widehat{k}]}}$	
Compute $\widehat{\mathbf{s}}, \mathbf{y}, \mathbf{F}, \mathbf{f}, f$ as in Equations (9) and (8) $e_0 \leftarrow \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$ $e_1 \leftarrow \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y} \bmod \widehat{q}\widehat{R}$ $t_0 \leftarrow \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$ $t_1 \leftarrow \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$		
	$\xrightarrow{(t_0, t_1)}$	
		$c \hookleftarrow U(\mathcal{C})$
	$\xleftarrow{c}$	
$\mathbf{z}_1 \leftarrow \mathbf{y}_1 + c\mathbf{s}_1$ $\perp$ **if** $\mathsf{Rej}_1(\mathbf{z}_1, c\mathbf{s}_1, \sigma_1, M_1) = 0$ $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + c\mathbf{s}_2$ $\perp$ **if** $\mathsf{Rej}_1(\mathbf{z}_2, c\mathbf{s}_2, \sigma_2, M_2) = 0$		
	$\xrightarrow{(\mathbf{z}_1, \mathbf{z}_2)}$	
		Checks Equations (11), (12), (13), and (14)

Fig. 7.1. Interactive zero-knowledge argument for commitment opening and user registration

where the first inequality holds by the probability preservation property of RD_∞, and the second inequality by Lemma 2.7. The exact same reasoning shows that the bounds on $\mathbf{z}_2$ and $\mathbf{z}_3^{\mathbb{Z}}$ are also verified with overwhelming probability. So

Equation (11) holds. Next, for all $i \in [\ell]$, we have

$$\begin{aligned}
\tau_0(h_i) &= \tau_0(g_i) + \sum_{j\in[256]} \gamma_{i,j} \mathbf{e}_j^{\mathbb{Z}^T} \left(\tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) - \mathbf{z}_3^{\mathbb{Z}}\right) \\
&\quad + \gamma_{i,257}\langle \tau(\mathbf{s}_1)\,,\ \tau(\mathbf{s}_1) - \mathbf{1}_{\widehat{n}m_1}\rangle \\
&= 0 + \sum_{j\in[256]} \gamma_{i,j}\cdot 0 + \gamma_{i,257}\cdot 0 \\
&= 0,
\end{aligned}$$

as desired, thus verifying Equation (12). Then, by definition of $\mathbf{z}_1, \mathbf{z}_2$, Equation (13) directly holds as $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = (\mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2) + c(\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2) = \mathbf{w} + c\mathbf{t}_A \bmod \widehat{q}\widehat{R}$. Finally, using the fact that $c^* = c$, we observe that $\mathbf{z} = \mathbf{y} + c\widehat{\mathbf{s}}$. We can thus compute

$$\begin{aligned}
\mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f &= e_0 + ce_1 + c^2(\widehat{\mathbf{s}}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\widehat{\mathbf{s}} + f) \\
&= e_0 + ce_1 \bmod \widehat{q}\widehat{R}.
\end{aligned}$$

Since we have $ct_1 - \mathbf{b}^T\mathbf{z}_2 + t_0 = e_0 + ce_1 \bmod \widehat{q}\widehat{R}$, it means that Equation (14) also holds. □

The proof of knowledge soundness of Lemma 7.2 follows the exact blueprint of that of [LNP22, Thm. B.7] and we thus do not include it.

Lemma 7.2. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i)\cdot(\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. We let $B = \sqrt{\widehat{n}m_1}$ be a bound on $\|\mathbf{s}_1\|_2$. Then, let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \alpha_1\eta B$, $\sigma_2 = \alpha_2\eta\sqrt{\widehat{n}m_2}$, $\sigma_3 = \alpha_3\sqrt{337}B$, and define $B_{256} = c_{256}\sigma_3\sqrt{256}$. Assume that $q_\pi > \max(B^2, 82/\sqrt{26}\cdot\widehat{n}m_1B_{256}, 2B_{256}^2/13 - B_{256})$.*

Then, the (interactive) zero-knowledge argument in Figure 7.1 is knowledge sound with an extractor running in expected polynomial time, and soundness error

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

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound for $\text{M-SIS}_{\widehat{n},\widehat{d},m_1+m_2,\widehat{q},\beta}$ for

$$\beta = 8\eta\sqrt{(c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1})^2 + (c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2})^2}$$

The zero-knowledge property follows from the M-LWE assumption (albeit in its knapsack form, which we detail in the proof) and the rejection sampling result. Although it is generally interesting to use the Rényi divergence that is provided in Lemma 2.8, its use for distinguishing problems such as this one is more delicate. In particular, as our situation does not satisfy the public sampleability property from [BLR+18], we are bound to use the statistical distance. As such, one needs to choose ε_i that are negligible in the security parameter. We give the proof for completeness.

Lemma 7.3. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i) \cdot (\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \alpha_1 \eta \sqrt{\widehat{n} m_1}$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$ and $\sigma_3 = \alpha_3 \sqrt{337}\sqrt{\widehat{n} m_1}$. We define $m_2' = \widehat{d} + 256/\widehat{n} + \ell + 1$ and assume that $m_2 > m_2'$. Then, the (interactive) zero-knowledge argument in Figure 7.1 is honest-verifier zero-knowledge. More precisely, there exists a simulator $\mathcal{S}$ that outputs a distribution that is ε-indistinguishable from that of an honest transcript, where*

$$\varepsilon = \frac{\varepsilon_1}{M_1} + \frac{\varepsilon_2}{M_2} + \frac{\varepsilon_3}{M_3} + 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1 - \delta_{q_{\min}}(m_2, m_2 - m_2')}$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{\widehat{n}, m_2 - (\widehat{d} + 256/\widehat{n} + \ell + 1), m_2, \widehat{q}, \chi}$*, and* $\delta_{q_{\min}}(\mathsf{a}, \mathsf{b}) = \mathbb{P}_{\mathbf{M} \sim U(\widehat{R}_{q_{\min}}^{\mathsf{b} \times \mathsf{a}})}[\mathbf{M} \cdot \widehat{R}_{q_{\min}}^{\mathsf{a}} \neq \widehat{R}_{q_{\min}}^{\mathsf{b}}]$ *is the singularity probability*[16].

Proof. We show by a sequence of hybrids how the transcript is simulatable without resorting to the secret $\mathbf{s}_1$. We first denote by $\mathcal{H}_0$ the distribution of an honest transcript. Late $\mathcal{H}_i$ denotes the distribution of the transcript in the protocol they describe.

$\underline{\mathcal{H}_1.}$ The prover performs the honest execution until it obtains the final challenge c from the verifier $\mathcal{V}$. It then rewinds $\mathcal{V}$'s inner randomness and restart the honest execution until it gets c'. If $c' \neq c$, it aborts, and else finishes the execution. Since $\mathcal{V}$ is an honest verifier, the challenges only depend on its inner randomness as they are sampled uniformly. As a result, because its inner randomness was rewound, it always holds that $c' = c$. This means that $\mathcal{H}_0$ and $\mathcal{H}_1$ are identically distributed.

$\underline{\mathcal{H}_2.}$ The prover $\mathcal{P}$ obtains c as in $\mathcal{H}_1$. On the following execution, after sampling $\mathbf{y}_1, \mathbf{y}_2$, it directly computes $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ and $\mathsf{keep}_i = \mathsf{Rej}_1(\mathbf{z}_i, c\mathbf{s}_i, \sigma_i, M_i)$. Instead of computing $\mathbf{w}$ as usual, it sets it as

$$\mathbf{w} \leftarrow \mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}_2 \mathbf{z}_2 - c\mathbf{t}_A \bmod \widehat{q}\widehat{R}.$$

It then proceeds as usual until it gets the challenges $(\mu_i)_{i \in [\ell + 2d\widehat{k}]}$. Instead of computing t_0 as usual, it sets it as

$$t_0 \leftarrow \mathbf{z}^T \mathbf{F} \mathbf{z} + c\mathbf{f}^T \mathbf{z} + c^2 f - (c\mathbf{t}_1 - \mathbf{b}^T \mathbf{z}_2) \bmod \widehat{q}\widehat{R},$$

where $\mathbf{z}$ is defined as in Equation (10) from $\mathbf{z}_1, \mathbf{z}_2$. It then aborts at this stage if $\mathsf{keep}_1 \wedge \mathsf{keep}_2 = 0$. By careful inspection, except for the new definition of $\mathbf{w}$ and t_0, the transcript is identical to that of the previous game. Regarding $\mathbf{w}$ and t_0, they are uniquely determined by all the other elements because of the verification of Equations (13) and (14). As such, the distribution of the transcript in $\mathcal{H}_2$ is identical to that from $\mathcal{H}_1$.

[16] This quantity is introduced in [BJRW23] and the authors show it can be bounded in our case by $\mathsf{b}\kappa \cdot q_{\min}^{-(\mathsf{a}-\mathsf{b}+1)n/\kappa}$. Since we already require $q_{\min}^{-n/\kappa} = \mathsf{negl}(\lambda)$ for soundness, it holds that all the $\delta_{q_{\min}}(\cdot, \cdot)$ are negligible.

$\underline{\mathcal{H}_3.}$ It proceeds exactly as in $\mathcal{H}_2$ except that it samples $\mathbf{z}_2$ directly from $\mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$, and sets $\mathsf{keep}_2 = 1$ with probability $1/M_2$ and 0 otherwise. By Lemma 2.8 and the data processing inequality of the statistical distance, it holds that $\Delta(\mathcal{H}_2, \mathcal{H}_3) \leq \frac{\varepsilon_2}{M_2}$.

$\underline{\mathcal{H}_4.}$ The execution is the same as $\mathcal{H}_3$ except in the way $\mathbf{t}_A, \mathbf{t}_B$ and t_1 are generated. At this step we invoke the hiding property of the ABDLOP commitment scheme where the commitment randomness is $\mathbf{s}_2$. Observe that $\mathbf{z}_2$ no longer depends on $\mathbf{s}_2$ because of the previous hybrid change. More precisely, in the initial commitment phase, it samples $\mathbf{u}_A$ from $U(\widehat{R}_{\widehat{q}}^{\widehat{d}})$, $\mathbf{u}_B$ from $U(\widehat{R}_{\widehat{q}}^{256/\widehat{n}+\ell})$, and u_1 from $U(\widehat{R}_{\widehat{q}})$. It then sets $\mathbf{t}_A = \mathbf{u}_A$, $\mathbf{t}_B = \mathbf{u}_B$. After receiving the challenges $(\mu_i)_i$, it sets $t_1 = u_1$.

The hiding property relies on $\begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}_{y,g} \\ \mathbf{b}^T\end{bmatrix}\mathbf{s}_2$ being pseudorandom, which holds based on the knapsack version of M-LWE. More precisely, it is based on the hardness of Knap-M-LWE$_{\widehat{n},m_2',m_2,\widehat{q},\chi}$ where $m_2' = \widehat{d}+256/\widehat{n}+\ell+1$. This knapsack version is proven as hard as the standard version of M-LWE in, e.g., [BJRW23, Lem. 4.1]. They show that there is a reduction from M-LWE$_{\widehat{n},m_2-m_2',m_2,\widehat{q},\chi}$ to the knapsack problem Knap-M-LWE$_{\widehat{n},m_2',m_2,\widehat{q},\chi}$ such that we can relate the hardness bounds of each problem as

$$\varepsilon_{\text{Knap-M-LWE}} \leq 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_{q_{\min}}(m_2, m_2 - m_2')},$$

where $\delta_{q_{\min}}(\mathsf{a},\mathsf{b}) = \mathbb{P}_{\mathbf{M}\sim U(\widehat{R}_{q_{\min}}^{\mathsf{b}\times\mathsf{a}})}[\mathbf{M}\cdot\widehat{R}_{q_{\min}}^{\mathsf{a}} \neq \widehat{R}_{q_{\min}}^{\mathsf{b}}]$ is the singularity probability. By [BJRW23, Lem. 2.6], the singularity probability can be bounded in our case as

$$\delta_{q_{\min}}(\mathsf{a},\mathsf{b}) < 1 - \prod_{i=0}^{\mathsf{b}-1}\left(1 - \frac{1}{q_{\min}^{(\mathsf{a}-i)\widehat{n}/\kappa}}\right)^{\kappa} \leq \mathsf{b}\kappa \cdot q_{\min}^{-(\mathsf{a}-\mathsf{b}+1)\widehat{n}/\kappa}.$$

Because we already require $q_{\min}^{-\widehat{n}/\kappa} = \mathsf{negl}(\lambda)$ for soundness, it holds that all the $\delta_{q_{\min}}(\cdot,\cdot)$ we consider are negligible. Since a distinguisher between $\mathcal{H}_3$ and $\mathcal{H}_4$ leads to a distinguisher for Knap-M-LWE, we have for any PPT adversary $\mathcal{A}$

$$\begin{aligned}|\mathbb{P}[\mathcal{A}(\mathcal{H}_3) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_4) = 1]| &\leq \varepsilon_{\text{Knap-M-LWE}} \\ &\leq 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_{q_{\min}}(m_2, m_2 - m_2')}.\end{aligned}$$

$\underline{\mathcal{H}_5.}$ It proceeds exactly as in $\mathcal{H}_4$ except that it samples $\mathbf{z}_1$ directly from $\mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$, and sets $\mathsf{keep}_1 = 1$ with probability $1/M_1$ and 0 otherwise. By Lemma 2.8 and the data processing inequality of the statistical distance, it holds that $\Delta(\mathcal{H}_4, \mathcal{H}_5) \leq \frac{\varepsilon_1}{M_1}$.

$\underline{\mathcal{H}_6.}$ It proceeds exactly as in $\mathcal{H}_5$ except that it samples the h_i directly from $U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\})$. Since the g_i are uniform in this set, and that the other terms also have a constant coefficient of zero, it holds that the h_i are identically distributed in $\mathcal{H}_5$ and $\mathcal{H}_6$. It yields that $\mathcal{H}_5$ and $\mathcal{H}_6$ are identical.

$\underline{\mathcal{H}_7.}$ It proceeds exactly as in $\mathcal{H}_6$ except that it samples $\mathbf{z}_3^{\mathbb{Z}}$ directly from $\mathcal{D}_{\mathbb{Z}^{256},\sigma_3}$ instead of $\mathbf{y}_3$, and sets $\mathsf{keep}_3 = 1$ with probability $1/M_3$ and 0 otherwise. By Lemma 2.8 and the data processing inequality of the statistical distance, it holds that $\Delta(\mathcal{H}_6, \mathcal{H}_7) \leq \frac{\varepsilon_1}{M_3}$.

Simulator $\mathcal{S}$. We now describe for completeness the simulator $\mathcal{S}$ which produces samples from $\mathcal{H}_7$ without resorting to the secret $\mathbf{s}_1$.

$\mathcal{S}$

```
$c \hookleftarrow U(\mathcal{C})$.
$\mathbf{t}_A \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}})$
$\mathbf{z}_1 \hookleftarrow \mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$
$\mathbf{z}_2 \hookleftarrow \mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$
$\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A \bmod \widehat{q}\widehat{R}$
$\mathbf{t}_B \hookleftarrow U(\widehat{R}_{\widehat{q}}^{256/\widehat{n}+\ell})$
$(\mathsf{R}_0, \mathsf{R}_1) \hookleftarrow U(\{0,1\}^{256\times\widehat{n}m_1})^2$
$\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$
$\mathbf{z}_3^{\mathbb{Z}} \hookleftarrow \mathcal{D}_{\mathbb{Z}^{256},\sigma_3}$
$\mathsf{keep}_3 = 1$ with prob. $1/M_3$
$(\gamma_{i,j})_{i,j} \hookleftarrow U(\mathbb{Z}_{q_\pi}^{\ell\times 257})$
$(h_i)_i \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^{\ell})$
$(\mu_i)_i \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\ell+2d\widehat{k}})$
$t_1 \hookleftarrow U(\widehat{R}_{\widehat{q}})$
Compute $\mathbf{F}, \mathbf{f}, f, \mathbf{z}$ from Equations (8) and (10)
$t_0 \leftarrow \mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f - (ct_1 - \mathbf{b}^T\mathbf{z}_2) \bmod \widehat{q}\widehat{R}$
$\mathsf{keep}_1 = 1$ with prob. $1/M_1$
$\mathsf{keep}_2 = 1$ with prob. $1/M_2$
if $\mathsf{keep}_3 = 0$ then
  return $((\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}), (\mathsf{R}_0, \mathsf{R}_1), \perp)$
elif $\mathsf{keep}_3 = 1$ and $\mathsf{keep}_1 \wedge \mathsf{keep}_2 = 0$ then
  return $((\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}), (\mathsf{R}_0, \mathsf{R}_1), \mathbf{z}_3^{\mathbb{Z}}, (\gamma_{i,j})_{i,j}, (h_i)_i, (\mu_i)_i, (t_0, t_1), c, \perp)$
else return $((\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}), (\mathsf{R}_0, \mathsf{R}_1), \mathbf{z}_3^{\mathbb{Z}}, (\gamma_{i,j})_{i,j}, (h_i)_i, (\mu_i)_i, (t_0, t_1), c,$
        $(\mathbf{z}_1, \mathbf{z}_2))$
```

The simulator $\mathcal{S}$ follows the construction of a transcript described in $\mathcal{H}_7$ in the honest-verifier setting. From the hybrids above, it holds that for any PPT adversary $\mathcal{A}$,

$$|\mathbb{P}[\mathcal{A}(\mathcal{H}_0) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{S}) = 1]| \leq \frac{\varepsilon_1}{M_1} + \frac{\varepsilon_2}{M_2} + \frac{\varepsilon_3}{M_3} + \varepsilon_{\text{Knap-M-LWE}},$$

as desired. □

7.2 Proof of Valid Credential

Algorithm 6.4 solely relies on a zero-knowledge argument for the signature verification of Algorithm 5.4. The user needs to hide its secret key, the desired attributes and the credential, while convincing the verifier that it holds such elements. We use the same techniques as in Section 7.1, although this relation is slightly more complex as it directly involves quadratic equations. Although we use the same notations, all the parameters of the proof system in this section (e.g. $m_1, m_2, q_1, \widehat{d}, \ell, \rho, \eta, \varepsilon_i, M_i$) are most likely different from those of the previous protocol unless specified otherwise.

Relation. The prover starts by reconstructing $\mathbf{v}_1$ as in Algorithm 5.4. For clarity, we denote by $\mathbf{m}_{\mathcal{I}}$ the sub-vector of attributes that are revealed and $\mathbf{m}_{sm}$ the sub-vector of concealed attributes concatenated with the secret key $\mathbf{s}$. We similarly define $\mathbf{D}_{\mathcal{I}}$ and $\mathbf{D}_{sm}$ such that $\mathbf{D}_s\mathbf{s} + \mathbf{D}\mathbf{m} = \mathbf{D}_{sm}\mathbf{m}_{sm} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}$. In particular, we let m_{sm} be the dimension of $\mathbf{m}_{sm}$, namely $m_{sm} = m_s + (m - |\mathcal{I}|)$. We use the same process to lift the relation modulo $\widehat{q} = q_1 q$ and to select the soundness amplification parameter ℓ, and the challenge space parameters k', ρ, η. The user proves knowledge of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathfrak{t}, \mathbf{m}_{sm}) \in R^{2d+kd+k+1+m_{sm}}$ such that

$$\begin{gathered} q_1\left(\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 + \mathbf{G}(\mathfrak{t}\mathbf{v}_2) - \mathbf{D}_{sm}\mathbf{m}_{sm}\right) = q_1\left(\mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}\right) \bmod \widehat{q}R \\ \|\mathbf{v}_1\|_2 \leq B_1', \|\mathbf{v}_2\|_2 \leq B_2, \|\mathbf{v}_3\|_2 \leq B_3 \\ \|\mathfrak{t}\|_2 = \sqrt{w}, \mathfrak{t} \in T_1, \mathbf{m}_{sm} \in T_1^{m_{sm}} \end{gathered}$$

where $\mathbf{A} \in R_q^{d \times 2d}$, $\mathbf{B}, \mathbf{A}_3 \in R_q^{d \times k}$, $\mathbf{G} \in R_q^{d \times kd}$, $\mathbf{D}_{sm} \in R_q^{d \times m_{sm}}$, $\mathbf{D}_{\mathcal{I}} \in R_q^{d \times |\mathcal{I}|}$, $\mathbf{u} \in R_q^d$, $\mathbf{m}_{\mathcal{I}} \in T_1^{|\mathcal{I}|}$, w, B_1', B_2 and B_3 are public elements part of the statement. We then embed everything using θ and M_θ. For clarity, we define $\mathbf{A}' = q_1 M_\theta(\mathbf{A})$, $\mathbf{B}' = q_1 M_\theta(\mathbf{B})$, $\mathbf{G}' = q_1 M_\theta(\mathbf{G})$, $\mathbf{A}_3' = q_1 M_\theta(\mathbf{A}_3)$, $\mathbf{D}_{sm}' = q_1 M_\theta(\mathbf{D}_{sm})$, and $\mathbf{u}' = q_1\theta(\mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}})$.

As it is needed later in the protocol, we detail how to tackle the quadratic term $\mathbf{G}'\theta(\mathfrak{t}\mathbf{v}_2)$, in particular how to express its i-th coefficient in terms of $\theta(\mathfrak{t})$ and $\theta(\mathbf{v}_2)$. Let $i \in [0, d\widehat{k}-1]$. We decompose it as $i = i_1\widehat{k}+i_2$ for $i_1 \in [0, d-1]$ and $i_2 \in [0, \widehat{k}-1]$. We call $\mathbf{e}_i$ the vector $\widehat{R}^{d\widehat{k}}$ that is 1 at position i and 0 elsewhere. We also call $\mathbf{e}_{i_2}$ the vector of $\widehat{R}^{\widehat{k}}$ that is 1 at position i_2 and 0 elsewhere. It holds that

$$[\theta(\mathfrak{t}\mathbf{G}\mathbf{v}_2)]_i = \mathbf{e}_i^T\theta((\mathbf{I}_d \otimes \mathfrak{t})\mathbf{G}\mathbf{v}_2) = \mathbf{e}_i^T(\mathbf{I}_d \otimes M_\theta(\mathfrak{t}))M_\theta(\mathbf{G})\theta(\mathbf{v}_2).$$

We have that $\mathbf{e}_i^T(\mathbf{I}_d \otimes M_\theta(\mathfrak{t})) = [\mathbf{0}_{1\times i_1\widehat{k}}|\mathbf{e}_{i_2}^T M_\theta(\mathfrak{t})|\mathbf{0}_{1\times(d-i_1-1)\widehat{k}}]$, where the non-zero block is at the block position i_1. We can now express

$$\mathbf{e}_{i_2}^T M_\theta(\mathfrak{t}) = \mathrm{Row}_{i_2}(M_\theta(\mathfrak{t})) = \theta(x^{\widehat{k}-1-i_2} \otimes_R \mathfrak{t})^T \cdot \mathbf{P} = \theta(\mathfrak{t})^T M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P},$$

where $\mathbf{P}$ is the permutation of $[0, \widehat{k}-1]$ having 1 only on the anti-diagonal, i.e.,

$$\mathbf{P} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

As a result, we have that $[\theta(\mathfrak{t}\mathbf{G}\mathbf{v}_2)]_i$ is equal to

$$\theta(\mathfrak{t})^T \cdot [\mathbf{0}_{\widehat{k}\times i_1\widehat{k}}|M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P}|\mathbf{0}_{\widehat{k}\times(d-i_1-1)\widehat{k}}]M_\theta(\mathbf{G}) \cdot \theta(\mathbf{v}_2),$$

which means the i-th coefficient of $\theta(q_1\mathbf{t}\mathbf{G}\mathbf{v}_2)$ can be expressed as $\theta(\mathbf{t})^T\mathbf{G}'_i\theta(\mathbf{v}_2)$, where

$$\mathbf{G}'_i = [\mathbf{0}_{\widehat{k}\times i_1\widehat{k}} | M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P}|\mathbf{0}_{\widehat{k}\times(d-i_1-1)\widehat{k}}]\mathbf{G}',$$

where the non-zero block is at position i_1, where $i = i_1\widehat{k}+i_2$ for $i_1 \in [0, d-1]$ and $i_2 \in [0, \widehat{k}-1]$. In the remainder of the protocol description, we define $\mathbf{v}'_j = \theta(\mathbf{v}_j)$ for $j \in [3]$, $\mathbf{t}' = \theta(\mathbf{t})$, $\mathbf{m}'_{sm} = \theta(\mathbf{m}_{sm})$.

The Protocol. We start by expressing ${B'_1}^2 - \|\mathbf{v}'_1\|_2^2$ as the sum of four square integer $a_{1,0}^2 + a_{1,1}^2 + a_{1,2}^2 + a_{1,3}^2$. Then, define $a_1 = a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3$ and $\mathbf{v}''_1 = [{\mathbf{v}'_1}^T|a_1]^T$ so that $\|\mathbf{v}''_1\|_2 = B'_1$. We perform the same decomposition and define $a_2, a_3, \mathbf{v}''_2, \mathbf{v}''_3$. We also define $\mathbf{A}'' = [\mathbf{A}'|\mathbf{0}_d]$, $\mathbf{B}'' = [\mathbf{B}'|\mathbf{0}_d]$, $\mathbf{G}''_i = [\mathbf{G}'_i|\mathbf{0}_{\widehat{k}}]$ and $\mathbf{A}''_3 = [\mathbf{A}'_3|\mathbf{0}_d]$. Later, we also pack the witnesses into the vector $\mathbf{s}_1 = (\mathbf{v}''_1, \mathbf{v}''_2, \mathbf{v}''_3, \mathbf{t}', \mathbf{m}'_{sm}) \in \widehat{R}^{m_1}$ for $m_1 = (2d\widehat{k}+1)+(kd\widehat{k}+1)+(k\widehat{k}+1)+\widehat{k}+m_{sm}\widehat{k}$.
<u>*First Round.*</u> We start by the main commitment phase which consists in committing to the witness, masks and randomness needed in subsequent rounds. We start by sampling $\mathbf{s}_2$ from χ^{m_2} where $Supp(\chi) \subseteq \widehat{S}_1$ and compute an Ajtai commitment of $\mathbf{s}_1$ with randomness $\mathbf{s}_2$ as $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$, where $\mathbf{A}_1 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_1})$ and $\mathbf{A}_2 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_2})$ are part of the common reference string crs. Then, we sample the Gaussian masks for what will later be $c\mathbf{s}_1$ and $c\mathbf{s}_2$. More precisely, we sample $\mathbf{y}_1$ from $\mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$ and $\mathbf{y}_2$ from $\mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$, and compute the commitment $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$.

We then sample a mask $\mathbf{y}_3$ from $\mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$ and a vector for soundness amplification by $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$ where all the entries are polynomials with a constant coefficient equal to zero. We later use $\widehat{\mathbf{m}}$ to denote the vector $\widehat{\mathbf{m}} = [\mathbf{y}_3^T|\mathbf{g}^T]^T \in \widehat{R}^{256/\widehat{n}+\ell}$. We commit to it via $\mathbf{t}_B = \mathbf{B}_{y,g}\mathbf{s}_2 + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$, where $\mathbf{B}_{y,g} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{(256/\widehat{n}+\ell)\times m_2})$ is part of crs.

The prover sends msg_1 as the first message and receives chal_1 as the first challenge, where they are both defined as

$$\begin{aligned}\mathsf{msg}_1 &= (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}) \in \widehat{R}_{\widehat{q}}^{2\widehat{d}+256/\widehat{n}+\ell}\\ \mathsf{chal}_1 &= \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) = (\mathsf{R}_0, \mathsf{R}_1) \in \left(\{0,1\}^{256\times m_1\widehat{n}}\right)^2\end{aligned}$$

with $(\mathsf{R}_0, \mathsf{R}_1)$ conditioned on $\|\mathsf{R}_0 - \mathsf{R}_1\|_2 \le \sqrt{337}$.
<u>*Second Round.*</u> We conclude the approximate range proof part. We define $\mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$. The challenge R is used to project the witness onto a smaller dimensional space and prove that the coefficients of $\mathbf{s}_1$ are small relative to $\widehat{q}$. In the second round, we respond to the challenge by masking $\mathsf{R}\tau(\mathbf{s}_1)$ with $\tau(\mathbf{y}_3)$. So we compute $\mathbf{z}_3^{\mathbb{Z}} = \tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) \in \mathbb{Z}^{256}$. Then, we perform rejection by sampling $u_3 \hookleftarrow U([0,1))$ and rejecting if

$$u_3 > \frac{1}{M_3}\exp\left(\pi\frac{-2\langle\mathbf{z}_3^{\mathbb{Z}}\ ,\ \mathsf{R}\tau(\mathbf{s}_1)\rangle + \|\mathsf{R}\tau(\mathbf{s}_1)\|_2^2}{\sigma_3^2}\right).$$

If the u_3 is smaller, then the prover accepts, sends msg_2 as the second message and receives chal_2 as the second challenge where they are defined by

$$
\begin{aligned}
&\mathsf{msg}_2 = \mathbf{z}_3^{\mathbb{Z}} \in \mathbb{Z}^{256} \\
&\mathsf{chal}_2 = \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) = (\gamma_{i,j})_{\substack{i\in[\ell] \\ j\in[262]}} \in \mathbb{Z}_{\widehat{q}}^{\ell\times 262}.
\end{aligned}
$$

Third Round. We now need to prove the equations over $\mathbb{Z}_{\widehat{q}}$, namely the ones related to norm constraints or binary vectors, as well as the well-formedness of $\mathbf{z}_3^{\mathbb{Z}}$.

$$
\begin{aligned}
&\tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) = \mathbf{z}_3^{\mathbb{Z}}, && (3.1b)\\
&\langle \tau(\mathbf{v}_1'')\,,\ \tau(\mathbf{v}_1'')\rangle = {B_1'}^2, && (3.2b)\\
&\langle \tau(\mathbf{v}_2'')\,,\ \tau(\mathbf{v}_2'')\rangle = B_2^2, && (3.3b)\\
&\langle \tau(\mathbf{v}_3'')\,,\ \tau(\mathbf{v}_3'')\rangle = B_3^2, && (3.4b)\\
&\langle \tau(\mathbf{t}')\,,\ \tau(\mathbf{t}')\rangle = w, && (3.5b)\\
&\langle \tau(\mathbf{t}')\,,\ \tau(\mathbf{t}') - \mathbf{1}_{\widehat{n}\widehat{k}}\rangle = 0. && (3.6b)\\
&\langle \tau(\mathbf{m}_{sm}')\,,\ \tau(\mathbf{m}_{sm}') - \mathbf{1}_{\widehat{n}\widehat{k}m_{sm}}\rangle = 0. && (3.7b)
\end{aligned}
$$

Using the same method and notations as in Section 7.1, we combine the quadratic equations with automorphisms over $\widehat{R}_{\widehat{q}}$ and define for all $i \in [\ell]$

$$
\begin{aligned}
h_i = g_i + \sum_{j\in[256]} &\gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{v}_1''^*\mathbf{v}_1'' - {B_1'}^2) \\
&+ \gamma_{i,258}(\mathbf{v}_2''^*\mathbf{v}_2'' - B_2^2) + \gamma_{i,259}(\mathbf{v}_3''^*\mathbf{v}_3'' - B_3^2) + \gamma_{i,260}(\boldsymbol{\mathsf{t}}'^*\boldsymbol{\mathsf{t}}' - w) \qquad (15)\\
&+ \gamma_{i,261}(\boldsymbol{\mathsf{t}}'^*(\boldsymbol{\mathsf{t}}' - \mathbf{1}_{\widehat{R}^{\widehat{k}}})) + \gamma_{i,262}({\mathbf{m}_{sm}'}^*(\mathbf{m}_{sm}' - \mathbf{1}_{\widehat{R}^{\widehat{k}m_{sm}}})).
\end{aligned}
$$

The prover then sends msg_3 as the third message and receives chal_3 as the third challenge, where they are both defined as

$$
\begin{aligned}
&\mathsf{msg}_3 = (h_1, \ldots, h_\ell) \in \widehat{R}_{\widehat{q}}^{\ell} \\
&\mathsf{chal}_3 = \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) = (\mu_i)_{i\in[\ell+d\widehat{k}]} \in \widehat{R}_{\widehat{q}}^{\ell+d\widehat{k}}.
\end{aligned}
$$

Fourth Round. All the quadratic equations, including the equations for the well-formedness of the h_i for the quadratic evaluations from the previous round, are then proven in round 4 directly over $\widehat{R}_{\widehat{q}}$. We need to prove that the h_i are well-formed and equal their expressions above, and we also need to prove the main quadratic relation $\mathbf{A}''\mathbf{v}_1'' - \mathbf{B}''\mathbf{v}_2'' + \mathbf{A}_3''\mathbf{v}_3'' - \mathbf{D}_{sm}''\mathbf{m}_{sm} + \mathbf{t}'\mathbf{G}''\mathbf{v}_2'' = \mathbf{u}'$. The latter represents $d\widehat{k}$ equations. We prove them all at once by combining them linearly with the challenges μ_i and prove that

$$\begin{aligned} 0 = \sum_{i \in [\ell]} \mu_i \Bigg(g_i &+ \sum_{j \in [256]} \gamma_{i,j} (\mathbf{e}_j^* \mathbf{y}_3 + \mathsf{r}_j^* \mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257} (\mathbf{v}_1''^* \mathbf{v}_1'' - {B_1'}^2) \\ &+ \gamma_{i,258} (\mathbf{v}_2''^* \mathbf{v}_2'' - B_2^2) + \gamma_{i,259} (\mathbf{v}_3''^* \mathbf{v}_3' - B_3^2) + \gamma_{i,260} (\mathbf{t}'^* \mathbf{t}' - w) \\ &+ \gamma_{i,261} (\mathbf{t}'^* (\mathbf{t}' - \mathbf{1}_{\widehat{R}^{\widehat{k}}})) + \gamma_{i,262} ({\mathbf{m}'_{sm}}^* (\mathbf{m}'_{sm} - \mathbf{1}_{\widehat{R}^{\widehat{k} m_{sm}}})) - h_i \Bigg) \\ + \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} (\mathbf{t}'^T \mathbf{G}_i'' \mathbf{v}_2'' &+ {\mathbf{e}_i^{\widehat{R}}}^T (\mathbf{A}'' \mathbf{v}_1'' - \mathbf{B}'' \mathbf{v}_2'' + \mathbf{A}_3'' \mathbf{v}_3'' - \mathbf{D}'_{sm} \mathbf{m}'_{sm} - \mathbf{u}')). \end{aligned} \tag{16}$$

For that let us define $\widehat{\mathbf{s}} = [\mathbf{s}_1^T | \mathbf{s}_1^* | \widehat{\mathbf{m}}^T | \widehat{\mathbf{m}}^*]^T$. We also define $\mathsf{r}_{1,j}$, $\mathsf{r}_{2,j}$, $\mathsf{r}_{3,j}$, $\mathsf{r}_{t,j}$, and $\mathsf{r}_{sm,j}$ such that

$$\mathsf{r}_j^* \mathbf{s}_1 = \mathsf{r}_{1,j}^* \mathbf{v}_1'' + \mathsf{r}_{2,j}^* \mathbf{v}_2'' + \mathsf{r}_{3,j}^* \mathbf{v}_3'' + \mathsf{r}_{t,j}^* \mathbf{t}' + \mathsf{r}_{sm,j}^* \mathbf{m}'_{sm}.$$

Then, the equation to be proven is equivalent to $\widehat{\mathbf{s}}^T \mathbf{F} \widehat{\mathbf{s}} + \mathbf{f}^T \widehat{\mathbf{s}} + f = 0 \bmod \widehat{q}\widehat{R}$, where

$$\begin{aligned} f = &- \sum_{i \in [\ell]} \mu_i \left(\sum_{j \in [256]} \gamma_{i,j} z_{3,j}^{\mathbb{Z}} + \gamma_{i,257} {B_1'}^2 + \gamma_{i,258} B_2^2 + \gamma_{i,259} B_3^2 + \gamma_{i,260} w + h_i \right) \\ &- \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} u_i' \end{aligned}$$

$$\mathbf{f} = \begin{bmatrix} \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_{1,j}^{*T} + \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} \mathbf{A}''^T \mathbf{e}_i^{\widehat{R}} \\ \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_{2,j}^{*T} - \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} \mathbf{B}''^T \mathbf{e}_i^{\widehat{R}} \\ \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_{3,j}^{*T} + \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} \mathbf{A}_3''^T \mathbf{e}_i^{\widehat{R}} \\ \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_{t,j}^{*T} \\ \sum_{i \in [\ell]} \sum_{j \in [256]} \mu_i \gamma_{i,j} \mathsf{r}_{sm,j}^{*T} + \sum_{i \in [d\widehat{k}]} \mu_{\ell+i} {\mathbf{D}'_{sm}}^T \mathbf{e}_i^{\widehat{R}} \\ \mathbf{0}_{2d\widehat{k}+1} \\ \mathbf{0}_{kd\widehat{k}+1} \\ \mathbf{0}_{k\widehat{k}+1} \\ -\sum_{i \in [\ell]} \mu_i \gamma_{i,261} \mathbf{1}_{\widehat{R}^{\widehat{k}}} \\ -\sum_{i \in [\ell]} \mu_i \gamma_{i,262} \mathbf{1}_{\widehat{R}^{\widehat{k} m_{sm}}} \\ \sum_{i \in [\ell]} \mu_i \sum_{j \in [256]} \gamma_{i,j} \mathbf{e}_j^{*T} \\ [\mu_1 | \ldots | \mu_\ell]^T \\ \mathbf{0}_{256/\widehat{n}} \\ \mathbf{0}_\ell \end{bmatrix} \tag{17}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}' & \mathbf{F}'' & \mathbf{0}_{m_1 \times 2(256/\widehat{n}+\ell)} \\ \multicolumn{3}{l}{\mathbf{0}_{m_1+2(256/\widehat{n}+\ell) \times 2(m_1+256/\widehat{n}+\ell)}} \end{bmatrix},$$

where

$$\mathbf{F}' = \begin{bmatrix} \mathbf{0}_{\widehat{k}(2d+kd+k)+3\times m_1} \\ \mathbf{0}_{\widehat{k}\times 2d\widehat{k}+1} \ \sum_{i\in[d]} \mu_{\ell+i}\mathbf{G}''_i \ \mathbf{0}_{\widehat{k}\times(m_1-d\widehat{k}(2+k)-2)} \\ \mathbf{0}_{\widehat{k}m_{sm}\times m_1} \end{bmatrix},$$

and

$$\mathbf{F}'' = \sum_{i\in[\ell]} \mu_i \cdot \mathrm{diag}(\gamma_{i,257}\mathbf{I}_{2d\widehat{k}+1}, \gamma_{i,258}\mathbf{I}_{kd\widehat{k}+1}, \gamma_{i,259}\mathbf{I}_{k\widehat{k}+1},$$
$$(\gamma_{i,260}+\gamma_{i,261})\mathbf{I}_{\widehat{k}}, \gamma_{i,262}\mathbf{I}_{\widehat{k}m_{sm}}).$$

Once we have defined these (public) matrices, we can compute the garbage terms and commit to them. More precisely, we define

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_1^{*T} \\ -\mathbf{B}_{y,g}\mathbf{y}_2 \\ -(\mathbf{B}_{y,g}\mathbf{y}_2)^{*T} \end{bmatrix} \in \widehat{R}_{\widehat{q}}^{2(m_1+256/\widehat{n}+\ell)}, \tag{18}$$

and compute $e_0 = \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$, $e_1 = \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y} \bmod \widehat{q}\widehat{R}$, and the commitments $t_0 = \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$ and $t_1 = \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$, where $\mathbf{b} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{m_2})$ is part of crs. The prover then sends msg_4 as the fourth message and receives chal_4 as the fourth challenge, where they are both defined as

$$\mathsf{msg}_4 = (t_0, t_1) \in \widehat{R}_{\widehat{q}}^2$$
$$\mathsf{chal}_4 = \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c \in \mathcal{C}.$$

Fifth Round. The final round consists in the final response that is typical of Schnorr-like proofs, i.e., outputting $z = y + c \cdot s$ without it leaking information. More precisely, the prover responds to the challenge by masking $c\mathbf{s}_1$ and $c\mathbf{s}_2$ with $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively. So we compute $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$. Then, we perform rejection by sampling $u_1, u_2 \hookleftarrow U([0,1))$ and rejecting if

$$u_1 > \frac{1}{M_1}\exp\left(\pi\frac{-2\langle\tau(\mathbf{z}_1)\ ,\ \tau(c\mathbf{s}_1)\rangle + \|\tau(c\mathbf{s}_1)\|_2^2}{\sigma_1^2}\right)$$
$$\text{or } u_2 > \frac{1}{M_2}\exp\left(\pi\frac{-2\langle\tau(\mathbf{z}_2)\ ,\ \tau(c\mathbf{s}_2)\rangle + \|\tau(c\mathbf{s}_2)\|_2^2}{\sigma_2^2}\right).$$

If the u_1, u_2 are both smaller than these respective bounds, then the prover accepts, sends msg_5 as the final message defined by

$$\mathsf{msg}_5 = (\mathbf{z}_1, \mathbf{z}_2) \in \widehat{R}^{m_1+m_2}.$$

Non-Interactive Proof. We summarize the proof and verification in Figure 7.2. The proof is $\pi = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$ as the elements $\mathbf{w}$ and t_0 and the challenges can be re-computed from the rest. The elements

$\mathbf{t}_A, \mathbf{t}_B, h_1, \ldots, h_\ell, t_1$ cannot be compressed as they all look uniformly random modulo q_π. We again use the entropy bound to evaluate the bit-size of discrete Gaussian vectors. It means the total bit-size can be bounded by

$$|\pi| \leq \left(\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 1\right) \widehat{n} \lceil \log_2 \widehat{q} \rceil + \widehat{n} m_1 (1/2 + \log_2 \sigma_1)$$
$$+ \widehat{n} m_2 (1/2 + \log_2 \sigma_2) + 256(1/2 + \log_2 \sigma_3) + \widehat{n} \lceil \log_2 (2\rho + 1) \rceil .$$

```
Proof
keep $\leftarrow 0$
while keep $= 0$ do
    $\mathbf{s}_2 \hookleftarrow \chi^{m_2}$
    $\mathbf{t}_A \leftarrow \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$
    $\mathbf{y}_i \hookleftarrow \mathcal{D}_{\widehat{R}^{m_i},\sigma_i}$ for $i \in \{1, 2\}$
    $\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$
    $\mathbf{y}_3 \hookleftarrow \mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$
    $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$
    $\mathbf{t}_B \leftarrow \mathbf{B}_{y,g}\mathbf{s}_2 + [\mathbf{y}_3^T|\mathbf{g}^T]^T \bmod \widehat{q}\widehat{R}$
    $\mathsf{msg}_1 \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w})$
    $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) \in (\{0,1\}^{256 \times m_1 \widehat{n}})^2$
    $\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$
    $\mathbf{z}_3^{\mathbb{Z}} \leftarrow \tau(\mathbf{y}_3) - \mathsf{R}\tau(\mathbf{s}_1)$
    $\mathsf{keep}_3 \leftarrow \mathsf{Rej}_1(\mathbf{z}_3^{\mathbb{Z}}, \mathsf{R}\tau(\mathbf{s}_1), \sigma_3, M_3)$
    $\mathsf{msg}_2 \leftarrow \mathbf{z}_3^{\mathbb{Z}}$
    $(\gamma_{i,j})_{i,j} \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) \in \mathbb{Z}_{\widehat{q}}^{\ell \times 262}$
    Compute all $h_i$ as in Equation (16)
    $\mathsf{msg}_3 \leftarrow (h_1, \ldots, h_\ell)$
    $(\mu_i)_{i \in [\ell + d\widehat{k}]} \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) \in \widehat{R}_{\widehat{q}}^{\ell + d\widehat{k}}$
    Compute $\widehat{\mathbf{s}}, \mathbf{y}, \mathbf{F}, \mathbf{f}, f$ as in Equations (18) and (17)
    $e_0 \leftarrow \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$
    $e_1 \leftarrow \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y} \bmod \widehat{q}\widehat{R}$
    $t_0 \leftarrow \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$
    $t_1 \leftarrow \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$
    $\mathsf{msg}_4 \leftarrow (t_0, t_1)$
    $c \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) \in \mathcal{C}$
    $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + c\mathbf{s}_1$
    $\mathsf{keep}_1 \leftarrow \mathsf{Rej}_1(\mathbf{z}_1, c\mathbf{s}_1, \sigma_1, M_1)$
    $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + c\mathbf{s}_2$
    $\mathsf{keep}_2 \leftarrow \mathsf{Rej}_1(\mathbf{z}_2, c\mathbf{s}_2, \sigma_2, M_2)$
    keep $\leftarrow \mathsf{keep}_1 \wedge \mathsf{keep}_2 \wedge \mathsf{keep}_3$
$\pi \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$
return $\pi$
```

```
Verification
$\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A \bmod \widehat{q}\widehat{R}$
$\mathsf{msg}_1 \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w})$
$(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1)$
$\mathsf{msg}_2 \leftarrow \mathbf{z}_3^{\mathbb{Z}}$
$(\gamma_{i,j})_{i,j} \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2)$
$\mathsf{msg}_3 \leftarrow (h_1, \ldots, h_\ell)$
$(\mu_i)_{i \in [\ell + d\widehat{k}]} \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3)$
Compute $\mathbf{F}, \mathbf{f}, f$ as in Equation (17)
$\mathbf{z} \leftarrow \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_1^{*T} \\ c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2 \\ (c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2)^{*T} \end{bmatrix}$
$t_0 \leftarrow \mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f - (ct_1 - \mathbf{b}^T\mathbf{z}_2) \bmod \widehat{q}\widehat{R}$
$\mathsf{msg}_4 = (t_0, t_1)$
$b_1 \leftarrow \|\mathbf{z}_1\|_2 \leq c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}$
$b_2 \leftarrow \|\mathbf{z}_2\|_2 \leq c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2}$
$b_3 \leftarrow \|\mathbf{z}_3^{\mathbb{Z}}\|_2 \leq c_{256}\sigma_3\sqrt{256}$
$b_4 \leftarrow \forall i \in [\ell], \tau_0(h_i) = 0$
$b_5 \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c$
return $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5$
```

Fig. 7.2. Non-interactive zero-knowledge argument for credential showing

Security Analysis. The proofs of Lemmas 7.4, 7.5 and 7.6 follow the same reasoning as that of Section 7.1. As the proof is presented to be non-interactive, there are a few modifications. In the completeness, the equations that would need to be satisfied on $\mathbf{w}$ and t_0 are automatically verified as these elements are recovered from c in the verification. Instead, one simply need to check that c indeed corresponds to the correct hash output. For knowledge soundness and zero-knowledge, the proof in the non-interactive case follows the same arguments as e.g. [BLNS23], which only slightly differs from the proofs of Section 7.1.

Lemma 7.4. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i) \cdot (\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \alpha_1 \eta B$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$ and $\sigma_3 = \alpha_3 \sqrt{337} B$. Then, the zero-knowledge argument in Figure 7.2 is complete.*

Lemma 7.5. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i) \cdot (\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Then, let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \alpha_1 \eta B$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$, $\sigma_3 = \alpha_3 \sqrt{337} B$, and define $B_{256} = c_{256} \sigma_3 \sqrt{256}$. Assume that $\widehat{q} > \max(B^2, 82/\sqrt{26} \cdot \widehat{n} m_1 B_{256}, 2B_{256}^2/13 - B_{256})$.*

Then, the zero-knowledge argument in Figure 7.2 is knowledge sound with an extractor running in expected polynomial time, and soundness error

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

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound for $\text{M-SIS}_{\widehat{n}, \widehat{d}, m_1 + m_2, \widehat{q}, \beta}$ for

$$\beta = 8\eta \sqrt{(c_{\widehat{n} m_1} \sigma_1 \sqrt{\widehat{n} m_1})^2 + (c_{\widehat{n} m_2} \sigma_2 \sqrt{\widehat{n} m_2})^2}$$

Lemma 7.6. *Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and M_1, M_2, M_3 in $(1, \infty)$. For $i \in [3]$, we define $\alpha_i = \sqrt{\pi}/\ln(M_i) \cdot (\sqrt{\ln(\varepsilon_i^{-1}) + \ln(M_i)} + \sqrt{\ln(\varepsilon_i^{-1})})$. We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Let χ be a distribution over S_1, and let $\sigma_1 = \alpha_1 \eta B$, $\sigma_2 = \alpha_2 \eta \sqrt{\widehat{n} m_2}$ and $\sigma_3 = \alpha_3 \sqrt{337} B$. We define $m_2' = \widehat{d} + 256/\widehat{n} + \ell + 1$ and assume that $m_2 > m_2'$. Then, the zero-knowledge argument in Figure 7.1 is zero-knowledge. More precisely, there exists a simulator $\mathcal{S}$ that outputs a distribution that is ε-indistinguishable from that of an honest proof, where*

$$\varepsilon = \frac{\varepsilon_1}{M_1} + \frac{\varepsilon_2}{M_2} + \frac{\varepsilon_3}{M_3} + 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1 - \delta_{q_{\min}}(m_2, m_2 - m_2')}$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{\widehat{n}, m_2 - (\widehat{d} + 256/\widehat{n} + \ell + 1), m_2, \widehat{q}, \chi}$, and $\delta_{q_{\min}}(\mathsf{a}, \mathsf{b}) = \mathbb{P}_{\mathbf{M} \sim U(\widehat{R}_{q_{\min}}^{\mathsf{b} \times \mathsf{a}})}[\mathbf{M} \cdot \widehat{R}_{q_{\min}}^{\mathsf{a}} \neq \widehat{R}_{q_{\min}}^{\mathsf{b}}]$ is the singularity probability.

8 Implementation and Performance

8.1 Parameter Selection and Estimated Performance

We now detail the parameter selection and evaluate the performance in terms of sizes that we expect for the signature and the proof systems. Our construction aims for a security target of $\lambda = 128$ bits of classical security. The security of the different algorithms and protocols rely on the structured lattice assumptions M-LWE and M-SIS. We estimate their hardness as described in Appendix A thus lower-bounding the security of our constructions. The security of our signature scheme however is set by a reduction from M-SIS with a loss described in Theorem 5.1 and 5.2. For a target of 128 bits, the security reduction entails a loss of around 38 bits for type ❶ forgeries and 41 bits for type ❷ forgeries for our parameters[17]. Concretely, this means we need to aim for 166 bits for the first M-SIS assumption and 169 bits for the second one. To meet this security requirement, we adjust the values of n, d, q and b, and set the rest according to Algorithm 5.1. For the proof systems, the parameters are set so as to minimize the proof sizes while providing sufficient security according to Lemmas 7.1 to 7.6. We also choose the rejection sampling parameters to achieve fewer rejections for a better computational efficiency at the expense of a slightly larger proof size. We suggest parameter sets for the signature scheme Table 8.4, and for the issuance and verification proof systems in Table 8.5 and 8.6 respectively, and summarize the estimated performance in Table 8.1. They correspond to a credential over 10 hidden attributes for comparison purposes with prior works, while also being the same order of magnitude as for the typical use case of identification documents, e.g., passport, national ID.

Security	Assumptions	$\lvert\mathsf{opk}\rvert$	$\lvert\mathsf{osk}\rvert$	$\lvert\mathsf{upk}\rvert$	$\lvert\mathsf{usk}\rvert$	$\lvert\mathsf{sig}\rvert$	$\lvert\mathsf{cred}\rvert$
128	M-SIS/M-LWE	47.53 KB	10 KB	2.38 KB	0.25 KB	6.81 KB	79.58 KB

Table 8.1. Size and security estimates of our anonymous credentials. All sizes are expressed in KB. $\mathsf{opk}, \mathsf{osk}$ represent the organization keys, $\mathsf{upk}, \mathsf{usk}$ the user keys, sig the emitted signature, and cred the credential proof.

We compare our scheme to the existing lattice-based anonymous credentials [JRS23,BLNS23,LLLW23] on their compromise between security and credential size, i.e., the size of a non-interactive proof in Algorithm 6.4, which represents the main metric we want to optimize over. First, in comparison to the figures reported in [JRS23], our construction drastically improves upon their performances on all metrics and with a tighter security proof. In particular, compared to their efficiency estimates in [JRS23, Tab. H.4], we gain factors of 200, 1080, 45, 9 on the organization public key, organization secret key, emitted

[17] Although we evaluate the composition $h^{\circ d}$ for our parameter selection, we explain in Appendix A how to bound $h^{\circ d}$ to support this loss.

signature, and credential proof respectively. We also achieve more compact sizes than [LLLW23]. In the latter, the authors propose parameter sets for three different security reductions. The first achieves credential proofs or around 190 KB but for selective unforgeability. The second builds upon the selective parameter set and achieves adaptive security via complexity leveraging for a credential of 370 KB, but it results in a reduction loss of 2^{128}. The third achieves adaptive security directly but results in much larger credentials of around 24.7 MB. Finally, in [BLNS23], the authors relaxed the hardness assumption by introducing the NTRU-ISIS$_f$ (and its interactive version) to reach smaller credentials. We reach credential proofs around 3 times smaller than their construction based on NTRU-ISIS$_f$, and get close to the performance of their construction based on the interactive assumption Int-NTRU-ISIS$_f$, but by relying on standard non-interactive assumptions (M-SIS, M-LWE). We summarize this comparison in Table 1.1. In particular, we are the only scheme achieving credentials smaller than 100 KB without relying on interactive and non-standard assumptions.

8.2 Implementation Details

To showcase the feasibility of our proposed construction, and to facilitate future research in this direction, we have implemented a proof of concept in C[18]. Apart from the complexity of the protocols themselves, the first notable challenge we faced was implementing polynomial arithmetic in *five* different rings, each presenting unique characteristics. Among these rings, three operate with coefficients modulo single-precision primes or single-precision products of two primes, posing challenges for efficient multiplication as they inherently lack native support for NTT. Another ring operates over multi-precision integers in order to estimate the spectral norm during the key generation whose methodology is described in Appendix B. The fifth ring is over $\mathbb{R}$ for the SEP perturbation sampling. We carried a precision analysis of the different Gaussian samplers in order to determine the necessary floating-point precision needed in the implementation of our scheme. It can be found in Section 9. Overall, we show that a precision of 53 bits is sufficient and leads to no noticeable security loss.

Faced with the intricacies of polynomial arithmetic across multiple rings, and considering that the actual construction is highly complex already[19], we chose FLINT [tea23] as our arithmetic backend. However, it is important to acknowledge several downsides of this choice: Firstly, FLINT implements arithmetic operations usually in a very generic way which may be non-optimal given that our parameters are static at compile time. Moreover, this generic arithmetic also includes the usage of branches for trivial cases, which breaks the constant-time paradigm for cryptographic implementations. Secondly, FLINT heavily relies on dynamic memory allocations, both internally and when handling passed data. In contrast to stack allocations, which are usually used in cryptographic imple-

[18] https://github.com/Chair-for-Security-Engineering/lattice-anonymous-credentials

[19] Excluding any arithmetic, our implementation has about 4700 lines of code compared to, e.g., 890 lines for the official Kyber code without arithmetic.

mentations, these dynamic ones are significantly slower. To mitigate this performance drop to a certain extent, we employ static, pre-allocated variables within the wrapper.

For these reasons, our implementation prioritizes accessibility and clarity for future research. We have abstracted calls to FLINT functions with a wrapper which offers the flexibility to replace the FLINT-based arithmetic with custom constant-time, parameter-specific code without the necessity of touching the protocol layer. Importantly, it requires no other dependencies beyond FLINT, and the code is thoroughly documented to enhance comprehension.

We want to emphasize that, apart from an parameter-specific, optimized arithmetic backend, our code could be further optimized by deploying AVX2-vectorized hashing. Through profiling, however, we have confirmed that for our code hashing is not the main bottleneck for both proof generation procedures as well as the verifications.

8.3 Implementation Performance

We benchmark our implementation on a laptop featuring an Intel Core i7 12800H CPU running at 4.6 GHz and the scaling governor set to `performance`. Both our code and the FLINT library have been compiled with `gcc 11.4.0` with the options `-O3 -march=native`. For building FLINT, we explicitly enabled AVX2 and disabled the `pthread` option to ensure that no thread pools are used and the program runs on a single core.

Protocol	Procedure	Time (ms)			
		min	mean	med	max
SEP	key gen.	241.01	414.21	270.33	1086.56
	sign	57.36	58.83	58.51	61.73
	verify	1.68	1.69	1.69	1.70
Credential Issuance	user commit	0.79	0.81	0.81	0.88
	signer sign cmt.	56.42	56.84	56.75	62.49
	user verify	1.68	1.69	1.69	1.76
	user key gen.	0.46	0.47	0.47	0.53
	user embed	0.74	0.78	0.78	0.86
	user prove	126.57	221.33	167.20	644.58
	signer verify	100.01	100.94	100.78	103.98
Credential Showing	user embed	2.35	2.39	2.38	2.52
	user prove	197.42	354.59	280.29	1019.18
	user verify	145.96	147.14	147.10	152.21

Table 8.2. Benchmark results. Statistics over 100 executions. Where applicable, the key and message were randomized (e.g., the SEP signing is benchmarked over random keys and random messages). High variance timings are due to rejection sampling. Note that we omitted the benchmark result for the oblivious signing user signature completion, which takes on average 1.2 µs.

Protocol	Procedure	Time (million cycles)			
		min	mean	med	max
SEP	key gen.	675.60	1161.12	757.79	3045.88
	sign	160.78	164.90	164.00	173.03
	verify	4.68	4.71	4.71	4.75
Credential Issuance	user commit	2.20	2.25	2.25	2.44
	signer sign cmt.	158.15	159.30	159.06	175.15
	user verify	4.69	4.72	4.71	4.91
	user key gen.	1.27	1.30	1.29	1.48
	user embed	2.07	2.17	2.17	2.40
	user prove	354.80	620.44	468.68	1806.93
	signer verify	280.32	282.93	282.50	291.48
Credential Showing	user embed	6.56	6.67	6.65	7.05
	user prove	553.41	993.99	785.72	2856.98
	user verify	409.15	412.46	412.32	426.67

Table 8.3. Benchmark results. Statistics over 100 executions. Where applicable, the key and message were randomized (e.g., the SEP signing is benchmarked over random keys and random messages). High variance timings are due to rejection sampling. Note that we omitted the benchmark result for the oblivious signing user signature completion, which takes on average 1611 cycles.

The timing results are shown in Table 8.2 in milliseconds, while the cycle counts are given in Table 8.3. As expected, there are notable variations in the timings due to rejection sampling, but also for procedures that do not involve rejection steps, which stems from the use of FLINT. Note, however, that we clear all FLINT-internal caches after each iteration of the benchmarked function.

The most important steps for anonymous credentials are Issuance and Credential Showing as they directly impact the user experience. Regarding Issuance, the full protocol takes about 400 ms (on average) which we deem very reasonable. Credential Showing is slightly slower as it takes about 500 ms (including Verification) on average, which should be imperceptible in most cases.

We also recall that the point of our implementation was to provide a better understanding of the performance of privacy-preserving solutions, not to provide the most optimized code for a specific setting. In particular, we did not implement our own arithmetic backend tailored to our moduli, nor did we leverage the multiple cores of modern CPUs (our timings were obtained without any parallelisation) or precomputations. In other words, there are many ways one could improve performance without changing the cryptographic protocol itself and, given the already appealing benchmarks as shown in Table 8.2, we are confident that our solution should be sufficiently practical for most use-cases.

9 Samplers' Precision Analysis

In this section, we detail the precision analysis of the different samplers that we require to determine the minimal floating-point precision for our implementation.

The systematic analysis of floating-point arithmetic (FPA) precision in Gaussian samplers has been bootstrapped by Prest and Lyubashevsky [LP15,Pre15,Pre17]. In these works, they provide a detailed floating-point precision analysis of Klein's sampler [Kle00,GPV08] and Peikert's sampler [Pei10].

Our construction uses three kinds of Gaussian samplers. The first is a spherical Gaussian sampler over $\mathbb{Z}$ (or $\mathbb{Z}^N$ or R), and is used as a base sampler for the others as well as for the masks in the zero-knowledge arguments. The second is the one presented in Algorithm 3.1 which samples a non-spherical perturbation over R. Finally, the third is the gadget sampler to sample points on $\mathcal{L}_q^\perp(\mathbf{G})$ where we use Klein's sampler on the basis (and its gram-Schmidt) of $\mathcal{L}_q^\perp(\mathbf{G})$. The base sampler is well understood and well studied which is why we only focus on the remaining two by assuming a perfect base sampler over $\mathbb{Z}$.

9.1 Klein's Sampler on the Gadget Lattice

The analysis of Klein's sampler has been thoroughly done in the general case by Prest [Pre15,Pre17]. We state the result in the case of integer bases and integer centers, and also specify how it changes when the (scaled) Gram-Schmidt is known exactly. Indeed, if the basis is integral, its Gram-Schmidt is rational and can be represented exactly if the denominators are not too large. The proof follows the blueprint of [Pre17] but we include it for completeness. We give the version of Klein's sampler that we use, which is rigorously equivalent to the standard formulation. In particular, it takes the basis $\mathbf{B} \in \mathbb{Z}^{N \times N}$, the scaled Gram-Schmidt $\widetilde{\mathbf{B}}'$ whose columns are the $\widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|_2^2$, the widths $s_i = s / \|\widetilde{\mathbf{b}}_i\|_2$ and a center $\mathbf{t} \in \mathbb{Z}^N$.

Algorithm 9.1: $\mathsf{Klein}(\mathbf{B}, \widetilde{\mathbf{B}}', (s_i)_i, \mathbf{c})$

Input: Basis $\mathbf{B} \in \mathbb{Z}^{N \times N}$, Scaled Gram-Schmidt $\widetilde{\mathbf{B}}' \in \mathbb{Q}^{N \times N}$, Widths $(s_i)_{i \in [N]} \in (\mathbb{R}^{+*})^N$, center $\mathbf{t} \in \mathbb{Z}^N$.

```
$\mathbf{v}_N \leftarrow \mathbf{0}$.
for $i = N, \ldots, 1$ do
    $d_i \leftarrow \langle \mathbf{t} - \mathbf{v}_i\ ,\ \widetilde{\mathbf{b}}'_i \rangle$.
    $z_i \hookleftarrow \mathcal{D}_{\mathbb{Z}, s_i, d_i}$.
    $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$.
```

Output: $\mathbf{v}_0$ ▷ Statistically close to $\mathcal{D}_{\mathbf{B}\mathbb{Z}^N, s, \mathbf{t}}$.

Lemma 9.1 ([Pre15, Lem. 3.12] adapted). *Let N be a positive integer, $\mathbf{B} \in \mathbb{Z}^{N \times N}$, $\mathbf{t} \in \mathbb{Z}^N$, $\varepsilon \in (0,1)$ and $s \geq 2\eta_\varepsilon(\mathbb{Z})\|\widetilde{\mathbf{B}}\|$. We let $\mathcal{P}$ be the output distribution of* $\mathsf{Klein}(\mathbf{B}, \widetilde{\mathbf{B}}', (s_i)_i, \mathbf{t})$ *with $\widetilde{\mathbf{B}}', (s_i)_i$ precomputed with infinite precision. Similarly, let $\overline{\mathcal{P}}$ be the output distribution of* $\mathsf{Klein}(\mathbf{B}, \overline{\widetilde{\mathbf{B}}'}, (\overline{s}_i)_i, \mathbf{t})$ *with $\overline{\widetilde{\mathbf{B}}'}, (\overline{s}_i)_i$ precomputed with finite precision. Let $\delta \in [0,1)$ be such that*

- $\forall i \in [N]$, $\|\overline{\widetilde{\mathbf{b}}'_i} - \widetilde{\mathbf{b}}'_i\|_\infty \leq \delta$
- $\forall i \in [N]$, $|s_i - \overline{s}_i| \leq \delta s_i$

We then define

$$\begin{aligned} C = \frac{\pi\delta N}{(1-\delta)^2} &\left(2c_N N\|\widetilde{\mathbf{B}}\| \left(c_N + \frac{(1+\delta)^2\varepsilon}{1-\varepsilon}\right)\right. \\ &\left. + (2+\delta)\left(c_N^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\right) + \frac{\pi\delta^2(1+(1+\delta)^2)c_N^2 N^3\|\widetilde{\mathbf{B}}\|^2}{(1-\delta)^2} \\ \underset{\delta,\varepsilon\to 0}{\sim}\ & \delta N(2\pi c_N^2 N\|\widetilde{\mathbf{B}}\| + 2\pi c_N^2 + 1). \end{aligned}$$

Then, it holds that for all $\mathbf{v} \in \mathbf{B}\mathbb{Z}^N$, $\overline{\mathcal{P}}(\mathbf{v}) \in [e^{-C}, e^{C}]\mathcal{P}(\mathbf{v})$. *When* $\widetilde{\mathbf{B}}'$ *can be computed exactly, the expression of* C *is improved to*

$$C = \delta N \frac{2\pi(1+\delta/2)}{(1-\delta)^2}\left(c_N^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right) \underset{\delta,\varepsilon\to 0}{\sim} \delta N(2\pi c_N^2 + 1).$$

Proof. Let $\mathbf{v} \in \mathbf{B}\mathbb{Z}^N$ be a possible outcome of $\mathcal{P}$ and $\overline{\mathcal{P}}$. There exists a unique $\mathbf{z} \in \mathbb{Z}^N$ such that $\mathbf{v} = \mathbf{B}\mathbf{z}$ whose entries are the outputs of the base sampler in the loop. In particular, there exists a unique $(d_i)_i$ (resp. $(\overline{d}_i)_i$) such that the infinite (resp. finite) precision sampler computes those centers in the loop.

We first bound the differences $\left|d_i - \overline{d}_i\right|$. Let $i \in [N]$. We can rewrite d_i in terms of $\mathbf{v}$ rather than $\mathbf{v}_i$ as $d_i = \langle \mathbf{t}-\mathbf{v}, \widetilde{\mathbf{b}}_i' \rangle + z_i$. Hence, $\overline{d}_i = \langle \mathbf{t}-\mathbf{v}, \widetilde{\mathbf{b}}_i' + \boldsymbol{\delta}_i \rangle + z_i = d_i + \langle \mathbf{t}-\mathbf{v}\ ,\ \boldsymbol{\delta}_i \rangle$, where by assumption $\|\boldsymbol{\delta}_i\|_\infty \leq \delta$. This gives

$$\left|d_i - \overline{d}_i\right| \leq \|\mathbf{t}-\mathbf{v}\|_2 \|\boldsymbol{\delta}_i\|_2 \leq c_N s\sqrt{N} \cdot \delta\sqrt{N} = c_N s\delta N,$$

where the last inequality comes from Lemma 2.7. Note that $d_i = \overline{d}_i$ if $\overline{\widetilde{\mathbf{B}}'}$ can be computed exactly.

Now, we bound the ratio $\mathcal{P}(\mathbf{v})/\overline{\mathcal{P}}(\mathbf{v})$. Since both sampler output $\mathbf{v}$ only if the base samplers output the z_i, we have

$$\frac{\mathcal{P}(\mathbf{v})}{\overline{\mathcal{P}}(\mathbf{v})} = \prod_{i\in[N]} \frac{\rho_{s_i,d_i}(z_i)\rho_{\overline{s}_i,\overline{d}_i}(\mathbb{Z})}{\rho_{\overline{s}_i,\overline{d}_i}(z_i)\rho_{s_i,d_i}(\mathbb{Z})} = \prod_{i=1}^{N} e^{u_i(z_i)} \frac{\rho_{\overline{s}_i,\overline{d}_i}(\mathbb{Z})}{\rho_{s_i,d_i}(\mathbb{Z})},$$

where $u_i(z) = \pi(z-\overline{d}_i)^2/\overline{s}_i^2 - \pi(z-d_i)^2/s_i^2$. By [Pre15, Lem. 3.10], we can bound the ratio of Gaussian sums and get

$$A := \sum_{i\in[N]} u_i(z_i) - \mathbb{E}_{z\sim\mathcal{D}_i}[u_i(z)] \leq \ln\frac{\mathcal{P}(\mathbf{v})}{\overline{\mathcal{P}}(\mathbf{v})} \leq \sum_{i\in[N]} u_i(z_i) - \mathbb{E}_{z\sim\overline{\mathcal{D}}_i}[u_i(z)] =: B,$$

where $\mathcal{D}_i = \mathcal{D}_{\mathbb{Z},s_i,d_i}$ and $\overline{\mathcal{D}}_i = \mathcal{D}_{\mathbb{Z},\overline{s}_i,\overline{d}_i}$. We now have to show that $-C \leq A$ and $B \leq C$.

First, we rewrite $u_i(z)$ in two different ways as in [Pre15]. We have

$$u_i(z) = \frac{\pi}{\overline{s}_i^2}((d_i - \overline{d}_i)^2 + 2(d_i - \overline{d}_i)(z - d_i) - \delta_i(2+\delta_i)(z-d_i)^2)$$
$$u_i(z) = \frac{\pi}{\overline{s}_i^2}(-(1+\delta_i)^2(d_i - \overline{d}_i)^2 + 2(1+\delta_i)^2(d_i - \overline{d}_i)(z - \overline{d}_i) - \delta_i(2+\delta_i)(z-\overline{d}_i)^2),$$

where $\overline{s}_i = (1+\delta_i)s_i$ with $|\delta_i| \leq \delta$ by assumption. We use the first expression and have the following inequalities

$$\begin{aligned}
|A| &\leq \sum_{i\in[N]} \frac{\pi}{\overline{s}_i^2}(2\left|d_i - \overline{d}_i\right|(|z_i - d_i| + |\mathbb{E}_{z\sim\mathcal{D}_i}[z-d_i]|) \\
&\qquad\qquad + \delta_i(2+\delta_i)((z_i - d_i)^2 + \mathbb{E}_{z\sim\mathcal{D}_i}[(z-d_i)^2])) \\
&\leq \frac{\pi}{(1-\delta)^2 s^2} \sum_{i\in[N]} 2c_N s\delta N\|\widetilde{\mathbf{b}}_i\|_2^2(|z_i - d_i| + |\mathbb{E}_{z\sim\mathcal{D}_i}[z-d_i]|) \\
&\qquad\qquad\qquad + \delta(2+\delta)\|\widetilde{\mathbf{b}}_i\|_2^2((z_i-d_i)^2 + \mathbb{E}_{z\sim\mathcal{D}_i}[(z-d_i)^2]) \\
&\leq \frac{2\pi c_N s N\delta\|\widetilde{\mathbf{B}}\|}{(1-\delta)^2 s^2}\left(\|\mathbf{v}-\mathbf{t}\|_1 + Ns\frac{\varepsilon}{1-\varepsilon}\right) \\
&\qquad + \frac{\pi\delta(2+\delta)}{(1-\delta)^2 s^2}\left(\|\mathbf{v}-\mathbf{t}\|_2^2 + Ns^2\left(\frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\right) \\
&\leq \frac{2\pi c_N N^2\delta\|\widetilde{\mathbf{B}}\|}{(1-\delta)^2}\left(c_N + \frac{\varepsilon}{1-\varepsilon}\right) + \frac{\pi\delta N(2+\delta)}{(1-\delta)^2}\left(c_N^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right) \\
&\leq C,
\end{aligned}$$

where the second inequality comes from the bound on $\left|d_i - \overline{d}_i\right|$ and the fact that $\overline{s}_i = (1+\delta_i)s/\|\widetilde{\mathbf{b}}_i\|_2 \geq (1-\delta)s/\|\widetilde{\mathbf{b}}_i\|_2$. The third inequality comes by bounding $\|\widetilde{\mathbf{b}}_i\|_2$ by $\|\widetilde{\mathbf{B}}\|$, by the fact that $\sum_i \|\widetilde{\mathbf{b}}_i\|_2|z_i - d_i| = \|\mathbf{v}-\mathbf{t}\|_1$, $\sum_i \|\widetilde{\mathbf{b}}_i^2\|_2(z_i - d_i)^2 = \|\mathbf{v}-\mathbf{t}\|_2^2$ and by bounding the expectations using [MR07, Lem. 4.2] as we have $s_i \geq 2\eta_\varepsilon(\mathbb{Z})$. The fourth inequality comes from the Gaussian tail bound of Lemma 2.7.

Following the method of [Pre15], we use the first expression of u_i for the $u_i(z_i)$ and the second expression for the expectations. Using the same arguments as before, we obtain

$$\begin{aligned}
|B| &\leq \frac{\pi\delta N}{(1-\delta)^2}\left(2c_N N\|\widetilde{\mathbf{B}}\|\left(c_N + \frac{(1+\delta)^2\varepsilon}{1-\varepsilon}\right) + (2+\delta)\left(c_N^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\right) \\
&\quad + \frac{\pi\delta^2(1+(1+\delta)^2)c_N^2 N^3\|\widetilde{\mathbf{B}}\|^2}{(1-\delta)^2} \\
&= C
\end{aligned}$$

The equivalence is taken at the first order in δ and ε which indeed simplifies to $\delta N(2\pi c_N^2 N\|\widetilde{\mathbf{B}}\| + 2\pi c_N^2 + 1)$.

Finally, the expression of C when $\widetilde{\mathbf{B}}'$ can be represented exactly comes from the exact same process. The only difference is that $d_i = \overline{d}_i$ which simplifies the two expressions of $u_i(z)$ to $u_i(z) = -\pi\delta_i(2+\delta_i)(z-d_i)^2/\overline{s}_i^2$. □

In our case, we apply Klein's sampler on the gadget lattice for centers which have integer coefficients. Since the gadget lattice has a specific structure, we can derive a closed-form expression of the scaled Gram-Schmidt which is why we may decide to store it exactly. More precisely in our case, $N = nkd$, $\|\widetilde{\mathbf{B}}\| = \sqrt{b^2+1}$. For typical parameters as those from Table 8.4 where $n = 256$, $k = 5$, $d = 4$, $b = 14$, we have $c_N \approx 0.453$ and therefore $C \approx 2^{28.82}\delta$ in the general case and $C \approx 2^{15.19}\delta$ in the exact scaled Gram-Schmidt case. Plugging this in our security proof gives a requirement of 50 bits and 36 bits of precision respectively aiming for $C = 1/2\sqrt{\lambda Q}$. The standard precision of 53 bits for floating points is therefore enough to incur almost no security loss.

9.2 Perturbation Sampler

The perturbation sampler is very similar to the Fast Fourier Sampler of [DP16] which is used in the Falcon signature scheme [PFH+20]. The algorithm is recursive in the subroutine that samples from $\mathcal{D}_{R,\sqrt{M_\tau(f_i)},d_i}$. In particular, it makes an overall number of $2d \cdot n$ calls to integer samplers $\mathcal{D}_{\mathbb{Z},s_j,e_j}$. We can, as is done for the Fast Fourier Sampler, analyze the precision needed for Algorithm 3.1 using an adapted version of the analysis of Klein's sampler. More precisely, we assume a relative error of at most δ_s on the s_j and an absolute error of at most δ_e on the centers e_j. We thus bound the quantities $|\overline{e}_j - e_j|$ by δ_e in the above proof, and the $|z_j - e_j|$ by $s_j t$ using [Lyu12, Lem. 4.4] for a slack $t \approx 6$. Using those upper bounds, and the fact that $s_j \geq \eta_\varepsilon(\mathbb{Z})$, we obtain that the relative error between the infinite and finite precision versions of the sampler is of $e^C - 1$, for

$$\begin{aligned} C = \frac{2\pi N}{(1-\delta_s)^2} & \left(\frac{\delta_e}{\eta_\varepsilon(\mathbb{Z})} \left(t + (1+\delta_s)^2 \frac{\varepsilon}{1-\varepsilon} \right) \right. \\ & \left. + \delta_s(1+\delta_s/2) \left(t^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon} \right) + \frac{\delta_e^2}{\eta_\varepsilon(\mathbb{Z})^2} \frac{1+(1+\delta_s)^2}{2} \right) \\ \leq N(2\pi t^2 & + 2\pi\sqrt{\varepsilon} + 1)(\delta_s + \delta_e), \end{aligned}$$

where $N = 2nd$, and where the inequality holds for all $\varepsilon, \delta_s, \delta_e \leq 2^{-10}$. In our context, this gives $C \leq 2^{18.83}(\delta_s + \delta_e)$, which when plugged into our security proof gives a precision requirement of $\delta_s + \delta_e \leq 2^{-39.4}$.

We use the same methodology than [PFH+20] to verify this bound. More precisely, we ran the signature process in both standard precision of 53 bits and high precision of 200 bits using the same random tape[20]. By comparing the values of the s_j and e_j between the two versions, we observe that we have $\delta_s + \delta_e \leq 2^{-36.9}$. Although this is slightly higher that $2^{-39.4}$, choosing the standard precision of 53 bits gives a sufficient margin so that it incurs no noticeable loss of security.

[20] Sampling can easily be made deterministic by generating the needed randomness via an extendable output function such as SHAKE256.

Conclusion

Practical signatures for privacy [CL04,PS16,San21] have been successfully implemented and even standardized [ISO13a,ISO13b], resulting in very efficient systems. Up until a few years ago, only theoretical alternatives existed in the post-quantum setting. Several recent works [JRS23,BLNS23,LLLW23] have improved the state-of-the-art by reducing the bandwidth, sometimes at the cost of a weaker security. We continue this line of works by showcasing promising sizes without compromising security but also extend it by demonstrating practicality of such mechanisms. Although we use anonymous credentials as a common benchmark with previous works, we stress that our SEP and its implementation are very versatile and could easily be adapted to other cryptographic primitives. Our work thus fosters practical post-quantum privacy and makes a significant step towards it.

Acknowledgments

Sven Argo, Tim Güneysu and Georg Land have been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972, and by the German Federal Ministry of Education and Research BMBF through the projects QuantumRISC (16KIS1038), PQC4Med (16KIS1044), and 6GEM (16KISK038). Corentin Jeudy and Adeline Roux-Langlois were supported by the French National Research Agency in the ASTRID program, under the national project AMIRAL with reference ANR-21-ASTR-0016, and Olivier Sanders through the MobiS5 project with reference ANR-18-CE-39-0019-02 MobiS5.

References

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.

BCR$^+$23. O. Blazy, C. Chevalier, G. Renaut, T. Ricosset, E. Sageloli, and H. Senet. Efficient Implementation of a Post-Quantum Anonymous Credential Protocol. In *ARES*, 2023.

BDGT17. A. Barki, N. Desmoulins, S. Gharout, and J. Traoré. Anonymous attestations made practical. In *WISEC*, 2017.

BEF19. D. Boneh, S. Eskandarian, and B. Fisch. Post-quantum EPID Signatures from Symmetric Primitives. In *CT-RSA*, 2019.

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.*, 2023.

BL07. E. Brickell and J. Li. Enhanced Privacy ID: a Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities. In *WPES*, 2007.

BLNS23. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In *CRYPTO*, 2023.

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.

BSZ05. M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA*, 2005.

Can20. R. Canetti. Universally Composable Security. *J. ACM*, 67(5):28:1–28:94, 2020.

CBC$^+$24. M. Christ, F. Baldimtsi, K. K. Chalkias, D. Maram, A. Roy, and J. Wang. SoK: Zero-Knowledge Range Proofs. *IACR Cryptol. ePrint Arch.*, page 430, 2024.

CDL16. J. Camenisch, M. Drijvers, and A. Lehmann. Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited. In *TRUST*, 2016.

CGM19. Y. Chen, N. Genise, and P. Mukherjee. Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures. In *ASIACRYPT*, 2019.

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

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

CKLL19. L. Chen, N. El Kassem, A. Lehmann, and V. Lyubashevsky. A Framework for Efficient Lattice-Based DAA. In *CYSARM@CCS*, 2019.

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.

DFPS22. J. Devevey, O. Fawzi, A. Passelègue, and D. Stehlé. On Rejection Sampling in Lyubashevsky's Signature Scheme. In *ASIACRYPT*, 2022.

DP06. C. Delerablée and D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *VIETCRYPT*, 2006.

DP16. L. Ducas and T. Prest. Fast Fourier Orthogonalization. In *ISSAC*, 2016.

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

EFG$^+$22. T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, and Y. Yu. Mitaka: A Simpler, Parallelizable, Maskable Variant of Falcon. In *EUROCRYPT*, 2022.

ETWY22. T. Espitau, M. Tibouchi, A. Wallet, and Y. Yu. Shorter Hash-and-Sign Lattice-Based Signatures. In *CRYPTO*, 2022.

FHS19. G. Fuchsbauer, C. Hanser, and D. Slamanig. Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials. *J. Cryptol.*, 2019.

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

GHL22. C. Gentry, S. Halevi, and V. Lyubashevsky. Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties. In *EUROCRYPT*, 2022.

GM18. Nicholas Genise and Daniele Micciancio. Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus. In *EUROCRYPT*, 2018.

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

GMR85. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In *STOC*, 1985.

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

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.

JHT22. H. Jia, Y. Hu, and C. Tang. Lattice-Based Hash-and-Sign Signatures using Approximate Trapdoor, Revisited. *IET Inf. Secur.*, 2022.

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

Kle00. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, 2000.

LLLW23. Q. Lai, F.-H. Liu, A. Lysyanskaya, and Z. Wang. Lattice-based Commit-Transferrable Signatures and Applications to Anonymous Credentials. *IACR Cryptol. ePrint Arch.*, page 766, 2023.

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

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

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

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

LP15. V. Lyubashevsky and T. Prest. Quadratic Time, Linear Space Algorithms for Gram-Schmidt Orthogonalization and Gaussian Sampling in Structured Lattices. In *EUROCRYPT*, 2015.

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

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.

LW15. V. Lyubashevsky and D. Wichs. Simple Lattice Trapdoor Sampling from a Broad Class of Distributions. In *PKC*, 2015.

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.

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

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

PFH$^+$20. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *FALCON. Tech. rep.*, 2020. Available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

Pre15. T. Prest. *Gaussian Sampling in Lattice-Based Cryptography.* PhD thesis, École Normale Supérieure, Paris, France, 2015.

Pre17. T. Prest. Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence. In *ASIACRYPT*, 2017.

PS15. D. Pointcheval and O. Sanders. Short Randomizable Signatures. *IACR Cryptol. ePrint Arch.*, page 525, 2015.

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

San21. O. Sanders. Improving Revocation for Group Signature with Redactable Signature. In *PKC*, 2021.

ST21. O. Sanders and J. Traoré. EPID with Malicious Revocation. In *CT-RSA*, 2021.

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

tea23. The FLINT team. *FLINT: Fast Library for Number Theory*, 2023. Version 3.0.0, https://flintlib.org.

Ver12. R. Vershynin. Introduction to the Non-Asymptotic Analysis of Random Matrices. In *Compressed Sensing*. 2012.

A Concrete Security Analysis

In this section we recall the methodology we use to estimate the hardness of the M-LWE and M-SIS assumptions. The best attacks use lattice reduction and rely on the BKZ algorithm as a subroutine. In our security estimates, we evaluate the cost of running the sieving SVP oracle with blocksize B by $2^{0.292B+16.4}$ for the classical security and $2^{0.257B+16.4}$ for the quantum security.

Under the Gaussian Heuristic and the Geometric Series Assumption, the BKZ algorithm with blocksize B would find a vector $\mathbf{v}$ in a N-dimensional lattice $\mathcal{L}$ with $\|\mathbf{v}\|_2 \leq \delta_B^N \mathrm{Vol}(\mathcal{L})^{1/N}$, where by [Che13] we have

$$\delta_B \approx \left(\frac{(\pi B)^{\frac{1}{B}} B}{2\pi e} \right)^{\frac{1}{2(B-1)}}. \tag{19}$$

A.1 Key Recovery and Zero-Knowledge: M-LWE

In the signature scheme of Section 5, the public key is given by $\mathbf{A}' \in R_q^{d \times d}$ and $\mathbf{B} = [\mathbf{I}_d | \mathbf{A}'] \mathbf{R} \bmod qR$ and the secret key by $\mathbf{R} \sim \mathcal{B}_1^{2d \times kd}$. Key recovery thus corresponds to an instance of search $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1}$ with kd binomial ternary secrets. We use the lattice estimator [APS15] on the instance $\text{LWE}_{nd,nd,q,\mathcal{B}_1}$ to determine the minimal BKZ block size B among all the evaluated attacks. We discard the structure of the underlying ring and simply extend the dimensions by the ring degree n by considering the matrix $M_\tau(\mathbf{A}')$. To account for the kd secrets, we consider the final cost to be that of running kd times BKZ which gives a cost of $kd2^{\nu B+16.4}$ for $\nu \in \{0.292, 0.257\}$.

For the zero-knowlegde property of the proof system, we evaluate the hardness of $\text{M-LWE}_{\widehat{n}, m_2-(\widehat{d}+256/\widehat{n}+\ell+1), m_2, \widehat{q}, \mathcal{B}_1}$ using the same method.

A.2 Forgery and Soundness: M-SIS

The complexity of the forgery and soundness are estimated either by the M-SIS assumption. To estimate the security of $\text{M-SIS}_{n,d,m,q,\beta}$, we find the cost of finding $\mathbf{v} \in \mathcal{L}_q^\perp([\mathbf{I}_d|\mathbf{A}])$ such that $\|\mathbf{v}\|_2 \leq \beta$ given $\mathbf{A} \sim U(R_q^{d \times m-d})$. We again look at the unstructured problem $\text{SIS}_{nd,nm,q,\beta}$. A standard optimization consists in finding a solution in a lattice of smaller dimension $nd \leq m^* \leq nm$ and completing the solution with zeros. We then use BKZ in block size B such that

$$\beta \geq \min_{nd \leq m^* \leq nm} \delta_B^{m^*} q^{nd/m^*}.$$

More precisely, for a fixed β, we find m^* that maximizes $\delta_B = \beta^{1/m^*} q^{-nd/m^{*2}}$ and then use Equation (19) to determine the corresponding block size B.

A.3 Bounding the Forgery Reduction Loss

The forgery reduction loss from Theorems 5.1 and 5.2 involves the d-th functional power of the function f, which stacks up the exponents $1 - 1/2\lambda$. It makes it slightly less intuitive to see why these d compositions do not deteriorate the reduction loss too much. For the sole sake of simplifying this intuition, we give the following bound on $h^{\circ d}$. We insist that this bound is used only to justify that the reduction is controlled despite the hybrid argument, but in the parameter selection we directly compute $h^{\circ d}$ and do not use this bound.

Lemma A.1. *Let a, b, c, μ be positive reals, and let α be in $(0, 1)$. We define the function h over $\mathbb{R}^+$ as*

$$h : x \in \mathbb{R}^+ \mapsto a + \mu \left(b + \mu(c + x)^{\alpha}\right)^{\alpha}.$$

Then, for all positive integer d, it holds that for all $x \geq 0$

$$\begin{aligned} h^{\circ d}(x) \leq \mu^{\frac{1}{1-\alpha}} \sum_{j\in[d]} &\left(\left(\mu^{\frac{-1}{1-\alpha}} a\right)^{\alpha^{2j-2}} + \left(\mu^{\frac{-1}{1-\alpha}} b\right)^{\alpha^{2j-1}} + \left(\mu^{\frac{-1}{1-\alpha}} c\right)^{\alpha^{2j}} \right) \\ &+ \mu^{\frac{1-\alpha^{2d}}{1-\alpha}} x^{\alpha^{2d}} \end{aligned}$$

Proof. We proceed by induction on d. For $d = 1$, we need to prove that $h(x) \leq \mu^{1/(1-\alpha)} \cdot ((\mu^{-1/(1-\alpha)}a) + (\mu^{-1/(1-\alpha)}b)^{\alpha} + (\mu^{-1/(1-\alpha)}c)^{\alpha^2}) + \mu^{(1-\alpha^2)/(1-\alpha)}x^{\alpha^2}$ which can be re-written as $h(x) \leq a + \mu b^{\alpha} + \mu^{1+\alpha}(c^{\alpha^2} + x^{\alpha^2})$. The inequality follows by the non-increasing property of p-norms for $p > 0$, that is $0 < p \leq q$ implies $\|\cdot\|_q \leq \|\cdot\|_p$. Here, we thus have $\|\cdot\|_1 \leq \|\cdot\|_{\alpha}$ as $\alpha < 1$, and thus $(\sum x_i)^{\alpha} \leq \sum x_i^{\alpha}$ for non-negative x_i. Hence, we get that for all $x \geq 0$

$$\begin{aligned} h(x) &\leq a + \mu(b^{\alpha} + (\mu(c + x)^{\alpha})^{\alpha}) \\ &\leq a + \mu(b^{\alpha} + \mu^{\alpha}(c + x)^{\alpha^2}) \\ &\leq a + \mu b^{\alpha} + \mu^{1+\alpha}(c^{\alpha^2} + x^{\alpha^2}). \end{aligned}$$

Now let us look at the induction step. Assume the inequality is verified at rank $d \geq 1$. Let $x \geq 0$. We have $h^{\circ(d+1)}(x) = h(h^{\circ d}(x))$. From the above, we get

$$h^{\circ(d+1)}(x) \leq a + \mu b^{\alpha} + \mu^{1+\alpha}c^{\alpha^2} + \mu^{1+\alpha}(h^{\circ d}(x))^{\alpha^2}.$$

Then, the induction hypothesis and the inequality $\|\cdot\|_1^{\alpha^2} \leq \|\cdot\|_{\alpha^2}^{\alpha^2}$ yields

$$\begin{aligned} h^{\circ d}(x)^{\alpha^2} \leq \mu^{\frac{\alpha^2}{1-\alpha}} \sum_{j\in[d]} &\left(\left(\mu^{\frac{-1}{1-\alpha}} a\right)^{\alpha^{2j}} + \left(\mu^{\frac{-1}{1-\alpha}} b\right)^{\alpha^{2j+1}} + \left(\mu^{\frac{-1}{1-\alpha}} c\right)^{\alpha^{2j+2}} \right) \\ &+ \mu^{\frac{\alpha^2-\alpha^{2d+2}}{1-\alpha}} x^{\alpha^{2d+2}}. \end{aligned}$$

Then, because $1 + \alpha + \alpha^2/(1-\alpha) = 1/(1-\alpha)$, and by reindexing the sum, we obtain

$$\begin{aligned} \mu^{1+\alpha} h^{\circ d}(x)^{\alpha^2} \leq \mu^{\frac{1}{1-\alpha}} \sum_{j\in[2,d+1]} &\left(\left(\mu^{\frac{-1}{1-\alpha}} a\right)^{\alpha^{2j-2}} + \left(\mu^{\frac{-1}{1-\alpha}} b\right)^{\alpha^{2j-1}} + \left(\mu^{\frac{-1}{1-\alpha}} c\right)^{\alpha^{2j}} \right) \\ &+ \mu^{\frac{1-\alpha^{2d+2}}{1-\alpha}} x^{\alpha^{2d+2}}. \end{aligned}$$

Finally, we observe that $a + \mu b^{\alpha} + \mu^{1+\alpha} c^{\alpha^2}$ is equal to the missing term $\mu^{1/(1-\alpha)} \cdot ((\mu^{-1/(1-\alpha)} a) + (\mu^{-1/(1-\alpha)} b)^{\alpha} + (\mu^{-1/(1-\alpha)} c)^{\alpha^2})$ which concludes the proof. □

Applying the above lemma for $\alpha = 1 - 1/2\lambda$, $a = c = k\varepsilon_{\text{M-LWE}}$, $b = 2a = 2k\varepsilon_{\text{M-LWE}}$ and $\mu = 5/4$, we can bound the corresponding loss terms from Theorems 5.1 and 5.2. The additive term depending can be bounded by $\varepsilon_+ = d\mu^{1/(1-\alpha)}((a/\mu^{1/(1-\alpha)})^{\alpha^{2d-1}} + (b/\mu^{1/(1-\alpha)})^{\alpha^{2d-2}} + (c/\mu^{1/(1-\alpha)})^{\alpha^{2d}})$ which in our case yields about a 12 bit loss compared to $\varepsilon_{\text{M-LWE}}$. We then have that $h^{\circ d}(C(|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}}) \leq \varepsilon_+ + 6(C(|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}})^{\alpha^{2d}}$. We can then plug this bound and obtain the required M-SIS hardness to achieve an advantage of $2^{-\lambda}$. In particular, for the parameters given in Table 8.4, we get that $\varepsilon_{\text{M-SIS}}$ should be smaller than $2^{-167.8}$ to ensure an advantage of at most 2^{-128} against type ❶ forgeries using the bounds we provide in this section. This is not far from the thorough parameter selection which gives a value of 2^{-166}. Doing the same for type ❷ forgeries would give $2^{-171.5}$ instead of 2^{-169}.

B Spectral Norm Estimation

During the key generation of the signature, we need to enforce a bound on the secret key, i.e., $\|M_\tau(\mathbf{R})\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{nkd} + 6)$, which requires the computation of $\|M_\tau(\mathbf{R})\|_2$. To avoid performing a singular value decomposition, we only approximate the value of $\|M_\tau(\mathbf{R})\|_2$. For that, we use the iterated power method, which we tweak to our specific use case. The iterated power method estimates the largest eigenvalue of a matrix $\mathbf{M}$ over $\mathbb{C}$ by selecting a random $\mathbf{u}$ over $\mathbb{C}$ and iterating ℓ times the update $\mathbf{u} \leftarrow \mathbf{M}\mathbf{u}/\|\mathbf{M}\mathbf{u}\|_2$ before returning $\mathbf{u}^H\mathbf{M}\mathbf{u}$ as the estimate of $\lambda_{\max}(\mathbf{M})$. The method is rather simple, but usually converges faster when $\mathbf{M}$ has separated eigenvalues, which is not the case of $M_\tau(\mathbf{R})M_\tau(\mathbf{R})^T$ where each eigenvalue is doubled by conjugation symmetry.

We thus change the approach to optimize this computation. First, we observe that $\|M_\tau(\mathbf{R})\|_2 = \max_{i\in[n]}\|\sigma_i(\mathbf{R})\|_2$ by [BJRW23, Lem. 2.3], where the σ_i are the complex embeddings of the field. As we work in cyclotomic fields, the conjugation symmetry allows to only look at $n/2$ embeddings. Hence, we have

$$\|M_\tau(\mathbf{R})\|_2 = \max_{i\in[n/2]}\|\sigma_i(\mathbf{R})\|_2 = \max_{i\in[n/2]}\sqrt{\lambda_{\max}(\sigma_i(\mathbf{R}\mathbf{R}^*))}.$$

We thus only have to estimate $n/2$ maximal eigenvalues of complex matrices with small dimensions ($\mathbb{C}^{2d\times 2d}$). For that we can update the iterated power method as follows. First, the updated vector $\mathbf{u}$ does not have to be re-normalize at each step, meaning that the estimate computes $\widetilde{\mathbf{u}} = \mathbf{M}^\ell\mathbf{u}$ and returns $\widetilde{\mathbf{u}}^H\mathbf{M}\widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|_2^2 = \mathbf{u}^H\mathbf{M}^{2\ell+1}\mathbf{u}/\mathbf{u}^H\mathbf{M}^{2\ell}\mathbf{u}$. Second, the starting vector $\mathbf{u}$ does not need to be random. In our experiments, choosing $\mathbf{u}$ to be the first column of $\mathbf{M}$ actually converges faster. In this case, the output value is

$$\frac{\mathbf{e}_1^T\mathbf{M}^{2\ell+3}\mathbf{e}_1}{\mathbf{e}_1^T\mathbf{M}^{2\ell+2}\mathbf{e}_1} = \frac{[\mathbf{M}^{2\ell+3}]_{1,1}}{[\mathbf{M}^{2\ell+2}]_{1,1}}.$$

Since $\mathbf{M}$ is some $\sigma_i(\mathbf{RR}^*)$, we have that

$$\|M_\tau(\mathbf{R})\|_2^2 \approx \max_{i\in[n/2]} \frac{\sigma_i([(\mathbf{RR}^*)^{2\ell+3}]_{1,1})}{\sigma_i([(\mathbf{RR}^*)^{2\ell+2}]_{1,1})}.$$

To minimize the number of matrix multiplications, we choose $\ell = 2^{\ell'} - 1$. As we need to compute $\mathbf{RR}^*$ to generate the perturbation sampling material, the spectral norm estimation thus requires $\ell' + 1$ matrix multiplication over $R^{2d\times 2d}$ to get $(\mathbf{RR}^*)^{2\ell+2}$, 1 extra multiplication to get $(\mathbf{RR}^*)^{2\ell+3}$ and then the computation of $2 \cdot n/2$ complex embeddings, i.e., two half FFT. In our implementation, we choose $\ell' = 4$ which gives the estimate

$$\|M_\tau(\mathbf{R})\|_2^2 \approx \max_{i\in[n/2]} \frac{\sigma_i([(\mathbf{RR}^*)^{33}]_{1,1})}{\sigma_i([(\mathbf{RR}^*)^{32}]_{1,1})}.$$

It approximates the actual norm with at least 10^{-5} precision, which is more than sufficient for our purposes. We note that although this estimate is rather fast, it requires computing $(\mathbf{RR}^*)^{33}$ in R and not R_q. As a result, the coefficients of $(\mathbf{RR}^*)^{33}$ become extremely large (around 420 bits) which calls for multi-precision integers. The renormalization in the iterated power method may mitigate this blow-up of coefficients but would require working over the complex embedded matrices directly. It in turn leads to more FFT computations (for all the matrix embeddings) and operations over floating-point complex numbers.

Symbol	Description	Value
	Signature Parameters	
λ	Security target	128
n	Signature ring degree	256
d	Module rank	4
m	Number of attributes	10
m_s	Secret dimension ($2d$)	8
q	Modulus	$425801 \approx 2^{18.7}$
k	Gadget length	5
b	Gadget base	14
ε	Smoothing loss for samplers	2^{-40}
s_1	Top preimage sampling width	5854.109
s_2	Bottom preimage sampling width	68.170
w	Hamming weight of tags	5
κ	Number of splitting factors of q	4
Q	Maximal number of signature queries	2^{32}
α	Rejection sampling slack (type ❷)	2.63997
M	Rejection sampling repetition rate (type ❷)	1.569
B_1'	First verification bound	128719.006
B_2	Second verification bound	2210.639
B_3	Third verification bound	1242.685
	Security Estimates	
$\lambda_{\text{❶}}$	Security target for M-SIS (type ❶)	166
$\lambda_{\text{❷}}$	Security target for M-SIS (type ❷)	169
λ_{KR}	Security target for M-LWE (key recovery)	152
$\beta_{\text{❶}}$	Euclidean bound for M-SIS (type ❶)	199463
$\beta_{\text{❷}}$	Euclidean bound for M-SIS (type ❷)	401099
$\mathrm{BKZ}_{\text{❶}}$	Required BKZ blocksize for M-SIS (type ❶)	653
$\mathrm{BKZ}_{\text{❷}}$	Required BKZ blocksize for M-SIS (type ❷)	560
BKZ_{KR}	Required BKZ blocksize for M-LWE (key recovery)	486
$\lambda^*_{\text{❶}}$	Reached M-SIS (classical) security (type ❶)	207
$\lambda^*_{\text{❷}}$	Reached M-SIS (classical) security (type ❷)	179
λ^*_{KR}	Reached M-LWE (classical) security (key recovery)	158
	Efficiency Estimates	
$\lvert\mathsf{pk}\rvert$	Size of public key ($\mathbf{B}$, rest generated from seeds)	47.5 KB
$\lvert\mathsf{sk}\rvert$	Size of secret key ($\mathbf{R}$)	10 KB
$\lvert\mathsf{sig}\rvert$	Size of signature ($\mathfrak{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3$)	6.81 KB

Table 8.4. Example parameter set for Anonymous Credentials (Signature Scheme).

Symbol	Description	Value
	Proof System Parameters	
λ	Security target	130
$\widehat{n}$	Proof system ring degree	64
$\widehat{k}$	Subring embedding dimension	4
$\widehat{d}$	Module rank	20
q_1	Modulus factor	$524201 \approx 2^{19}$
$q_{\min}$	Smallest modulus factor	425801
$\widehat{q}$	Proof system modulus (qq_1)	223205310001
ℓ	Soundness amplification dimension	7
$\lvert\mathcal{I}\rvert$	Number of disclosed attributes	0
m_1	Witness dimension	104
m_2	Dimension of ABDLOP randomness	58
χ	Distribution of ABDLOP randomness	$\mathcal{B}_1$
ρ	Infinity norm of challenges	8
η	Manhattan-like norm of challenges	93
$(\alpha_1, \alpha_2, \alpha_3)$	Rejection sampling slacks	$(48.64, 48.64, 48.64)$
(M_1, M_2, M_3)	Rejection sampling repetition rates	$(2, 2, 2)$
σ_1	First rejection sampling width	369050.897
σ_2	Second rejection sampling width	275602.779
σ_3	Third rejection sampling width	72848.106
	Security Estimates	
β	Euclidean bound for M-SIS	11551631225.350
$\text{BKZ}_{\text{M-SIS}}$	Required BKZ blocksize for M-SIS	395
$\text{BKZ}_{\text{M-LWE}}$	Required BKZ blocksize for M-LWE	386
$\lambda^*_{\text{M-SIS}}$	Reached M-SIS (classical) security	131
$\lambda^*_{\text{M-LWE}}$	Reached M-LWE (classical) security	129
δ	Soundness error	$2^{-128.26}$
	Efficiency Estimates	
$\lvert\pi\rvert$	Verification proof size	35.99 KB

Table 8.5. Example parameter set for Anonymous Credentials (Proof System) for the issuance proof.

Symbol	Description	Value
	Proof System Parameters	
λ	Security target	128
$\widehat{n}$	Proof system ring degree	64
$\widehat{k}$	Subring embedding dimension	4
$\widehat{d}$	Module rank	23
q_1	Modulus factor	$549755813881 \approx 2^{39}$
$q_{\min}$	Smallest modulus factor	425801
$\widehat{q}$	Proof system modulus (qq_1)	234086575306343681
ℓ	Soundness amplification dimension	7
$\lvert\mathcal{I}\rvert$	Number of disclosed attributes	0
m_1	Witness dimension	211
m_2	Dimension of ABDLOP randomness	74
χ	Distribution of ABDLOP randomness	$\mathcal{B}_1$
ρ	Infinity norm of challenges	8
η	Manhattan-like norm of challenges	93
$(\alpha_1, \alpha_2, \alpha_3)$	Rejection sampling slacks	$(48.64, 48.64, 48.64)$
(M_1, M_2, M_3)	Rejection sampling repetition rates	$(2, 2, 2)$
σ_1	First rejection sampling width	582380223.293
σ_2	Second rejection sampling width	311304.541
σ_3	Third rejection sampling width	114957846.739
	Security Estimates	
β	Euclidean bound for M-SIS	21756342921843.957
$\text{BKZ}_{\text{M-SIS}}$	Required BKZ blocksize for M-SIS	396
$\text{BKZ}_{\text{M-LWE}}$	Required BKZ blocksize for M-LWE	382
$\lambda^*_{\text{M-SIS}}$	Reached M-SIS (classical) security	132
$\lambda^*_{\text{M-LWE}}$	Reached M-LWE (classical) security	127
δ	Soundness error	$2^{-128.37}$
	Efficiency Estimates	
$\lvert\pi\rvert$	Verification proof size	79.58 KB

Table 8.6. Example parameter set for Anonymous Credentials (Proof System) for the verification proof.