

Compact Lattice Anonymous Credentials from Tighter Approximate Range Proofs

Corentin Jeudy[✉] and Olivier Sanders[✉]

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

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

Abstract. Accommodating cryptographic authenticity with strong user privacy assurances has been the primary motivation for anonymous credentials systems. Their features have recently come into the spotlight with the European Digital Identity (EUDI) wallet initiative, insisting on the need for efficient and private solutions based on well-understood security foundations for high assurances. This coincides with the post-quantum transition, but current quantum-safe solutions based on standard assumptions are still lagging behind the ones on ad-hoc interactive assumptions performance-wise.

In this paper, we present several techniques to improve the efficiency of anonymous credentials from standard lattice assumptions, narrowing the gap with more efficient but also more exotic ones. Alongside other optimizations, our main contributions focus on the zero-knowledge protocol of Lyubashevsky, Nguyen, Plançon (Crypto’22), currently the efficiency bottleneck of lattice privacy-oriented constructions, for which we identify several sources of improvements that may be of independent interest.

Keywords: Lattice-Based Cryptography · Anonymous Credentials · Privacy · Zero-Knowledge Proof

1 Introduction

An anonymous credential (AC) system allows users to request credentials on a set of attributes from an organization (i.e., an issuer) and then to use them to attest some rights to verifiers with very high privacy assurances. A typical example is the one where the user is a citizen requesting a credential attesting their attributes (e.g. name, birth date, address, city of residence, etc.) from a government agency (playing the role of the organization) and then uses it to prove to some service provider (playing the role of the verifier) that, e.g., their city of residence is indeed the one they claim. Using digital signatures would be possible here, but would require to reveal the full list of attributes to the verifier which is clearly undesirable. The strength of anonymous credentials is that they provide this user complete control of the disclosed information. They may decide to reveal only a subset of their attributes to a verifier $\mathcal{V}_1$ and a totally different subset to a second verifier $\mathcal{V}_2$. These two presentations remain unlinkable, unless the revealed attributes trivially enable the verifiers to associate them. Better

still, these privacy features also hold with respect to the organization that cannot trace the issued credentials unless, here again, the revealed attributes inherently allow it to do so. Similar privacy requirements may also be formulated for the issuance process where the user can decide to hide some of the attributes to the organization when receiving its credential.

After decades of academic scrutiny, anonymous credentials seemed to be promised to a bright industrial future with the recent EUDI (European Digital Identity) regulation [otEU24] that lists the requirements which shall be fulfilled by the future EUDI wallet. Indeed, this regulation requires that authentication should ensure “selective disclosure of data”, “enable privacy preserving techniques which ensure unlinkability”, and that organizations should not be able “to obtain data that allows transactions or user behaviour to be tracked, linked or correlated”. For readers familiar with AC, this looks like the legal transposition of the AC security model. In parallel, constructions of pre-quantum anonymous credentials have proved extremely efficient (e.g. [ASM06,PS16,TZ23]) and are thus more than able to comply with real-world functional requirements.

Yet, the European Commission (EC) has so far favored alternative solutions, which incited many cryptographers to publish a document [BBC+24] strongly encouraging the EC to reconsider its position. Unfortunately, the latter is driven by a very technical constraint. To avoid cloning, credentials should be linked to a secure element, a property usually referred to as device-binding. While there is no obstacle in theory, this conflicts in practice with the design choices made by the cryptographic community that focused, for performance reasons, on pairing-based constructions, proven under non-standard assumptions. Unfortunately, national cybersecurity certification authorities have so far been reluctant to approve such solutions, which explains why they are not listed in the so-called SOG-IS documents [Sec23] and its successor, the Agreed Cryptographic Mechanisms by ENISA [ENI26]. As a consequence, existing secure elements do not support pre-quantum anonymous credentials, which is a strong challenge to address when designing a real-world solution (see [TD24,Leh26] for more details on the context).

This momentum for private authentication coincides with the post-quantum migration that is urged by most cybersecurity agencies worldwide. While the celebrated NIST competition did not consider anonymous credentials systems, several post-quantum constructions, e.g., [JRS23,BLNS23,AGJ+24,JS26], were recently published, with more and more attractive performance. As of writing, we note that we can classify them into two main categories. Those proven under standard lattice assumptions (M-SIS, M-LWE), where presentations are about 47 KB large [JS26], and those proven under interactive, ad-hoc assumptions with presentation size below 30 KB [BLNS23]. While performance is obviously very important for real-world systems, the pre-quantum experience shows that it is not sufficient. In particular, the ability to rely on standard assumptions is at least as important to get support from national cybersecurity agencies and thus to be deployed in secure elements. In this work, we then focus on anonymous credentials proven under standard lattice assumptions and aim at improving their

performance, to close the gap with constructions based on interactive assumptions.

Current hurdles. Before presenting our contributions, we just recall that most anonymous credentials follow a rather simple framework. A credential is a signature generated by the organization on a set of (potentially concealed) attributes during the issuance phase. When presenting the credential, the user essentially produces a zero-knowledge proof of knowledge of this signature that convinces the verifier while ensuring anonymity. The key to performance is to associate the right signature and zero-knowledge framework, so as to ensure smooth interactions between these two crucial building blocks. Arguably, the recent works have so far focused on the signature component, to reduce its size and thereby that of the overall construction. It proved fruitful, but we note that it has led to a clear imbalance, with a zero-knowledge proof that is by far the largest component. For example, the construction in [JS26] yields credentials of only 3.52 KB (which is not much larger than ML-DSA) but presentation proofs of 47 KB. We therefore believe that it is time to focus on the zero-knowledge component, which is instantiated with the powerful LNP framework [LNP22] in most cases.

The key feature of the LNP framework is its ability to prove relations of the form $f(\mathbf{w}) = 0 \bmod q\mathcal{R}$, for some modulus q and quadratic function f , along with $\|\mathbf{w}\|_2 \leq B$, which covers the need of many privacy-preserving applications. Without going into the technicalities of it, the LNP framework starts by computing an extension of the Ajtai commitment of $\mathbf{w}$ and then proves that the committed values satisfy all the relations above. This however raises a very practical concern: this commitment must be computed in some quotient ring $\mathcal{R}_{q_\pi}$, which means that the relations we actually prove hold modulo q_π . The main challenge then is to transpose this argument (valid modulo q_π) to relations in $\mathcal{R}_q$, and even $\mathbb{Z}$ in the case of the norm relation $\|\mathbf{w}\|_2 \leq B$. One could try to simplify this challenge by choosing a modulus q dividing q_π but this may create a series of problems in practice. For example, one may want to select a small q for efficiency or security reasons (e.g. in the context of LWE encryption), but this leads q_π to have a small factor, which harms soundness of the LNP proofs. One must then compensate by adding new elements in these proofs, which affects the overall performance. Similarly, the LNP proof imposes certain arithmetic constraints on q_π , due to the challenge space $\mathcal{C}$, which may conflict with other requirements on q that comes from another system. In the end, this condition “ q divides q_π ” may lead to either selecting a q that is not optimal for the targeted application or to further complexifying the LNP proof.

More concretely, the challenge above is that we have proven some statement $P(\mathbf{w}) = q\mathbf{j} \bmod q_\pi\mathcal{R}$, potentially with $\mathbf{j} = 0$ (for relations in $\mathbb{Z}$), and we want to prove that it remains true without the “ $\bmod q_\pi\mathcal{R}$ ”. To address this problem, the authors of [LNP22] resort to approximate range proofs based on Johnson-Lindenstrauss (JL) heuristics. In a few words, the latter allow for deriving a bound on the norm of $\mathbf{x}$ based on that of $\mathbf{z} = \mathbf{y} + \mathbf{R}\mathbf{x} \bmod q_\pi$, where $\mathbf{R}$ has small random coefficients and $\mathbf{y}$ is some mask. By properly selecting $\mathbf{y}$, one can

derive an approximate bound B_{arp} on $\mathbf{x}$ while hiding it, which is essential for retaining the zero-knowledge property. In our case, we can for example set $\mathbf{x}$ as either $\mathbf{w}$ or $\mathbf{j}$, or even their concatenation. Thanks to this approximate bound B_{arp} , one can now derive an approximate bound on $P(\mathbf{w}) - q\mathbf{j}$ and then select a modulus q_π that is large enough to ensure that no wrap-around occurs. In other words, we are ensured that the relations we prove modulo q_π actually hold in $\mathbb{Z}$, and in turn modulo q when relevant, which concludes the proof from the security standpoint.

From the efficiency standpoint, the situation is however far less satisfying. The bound on $P(\mathbf{w})$ is the one obtained from its evaluation in $\mathbb{Z}$ (without any reduction), which can be quite large, especially for quadratic P , and we cannot do better than qB_{arp} for bounding $q\mathbf{j}$, which can be even bigger. Moreover, we stress that B_{arp} is in practice much larger than the actual norms of $\mathbf{w}$ and $\mathbf{j}$ as it is derived from a vector $\mathbf{z}$ whose norm is significantly larger than the one of $\mathbf{R}\mathbf{x}$ (because of the masking term $\mathbf{y}$) which is itself larger than $\|\mathbf{x}\|$. Finally, the approximate range proofs in [LNP22] are based on a series of technical lemmas that add other constraints on q_π , e.g., q_π must be larger than $41mB_{arp}$ where m is the dimension of approximate range proof witness $\mathbf{x}$. All in all, it results in a modulus q_π which can be very large (e.g. 2^{65} in the very recent construction in [JS26]), which directly impacts overall performance.

The LNP proof additionally involves so-called garbage polynomials $\mathbf{g}$ to prove quadratic evaluations like integer relations. This is typically done by embedding the integer relation within the constant coefficient of a polynomial, and performing operations directly on the entire polynomial. The purpose of $\mathbf{g}$ is then to blind the coefficients that are irrelevant for the relations we prove but that could still leak information on the underlying witnesses, e.g., all but the constant one. As $\mathbf{g}$ cannot be revealed and must not be chosen after seeing the verifier’s challenge c , it is committed to in the first steps of the protocol, which is not efficient as most of its coefficients are uniform elements modulo q_π . In [LNP22], the authors rely on an extension of the BDLOP hiding commitment scheme [BDL⁺18] that can deal with large elements but at some cost, as its size linearly grows with the number of such elements. Although the resulting commitment is not the largest part of the LNP proof, it still adds an overhead that is not insignificant in practice.

1.1 Our Contribution

In this work, we improve the performance of anonymous credentials proven under standard assumptions, mostly by introducing several ways of improving the LNP proof system. While most of our contributions target the AC constructions (e.g. [AGJ⁺24,JS26]) based on the gadget samplers [MP12,JS25b,JS26], we note that they are generic enough to be transposed to similar contexts with few efforts.

Relaxed Requirement for Lifting Proofs. Our first contribution starts with the observation that the authentication mechanisms based on gadget sam-

plers [MP12, JS25b, JS26], for example [dPLS18, AGJ⁺24, JS25a, JS26], share a similar structure. The signer usually starts with some public matrix $\mathbf{A}'$ without any specific property, from which it generates the remainder of its public key. The signature is then a short vector $\mathbf{v}$ satisfying some relation $[\mathbf{A}'|\mathbf{B}]\mathbf{v} = \mathbf{0} \bmod q$, where $\mathbf{B}$ is some public matrix that we leave unspecified. When proving unforgeability, the reduction selects the matrix $\mathbf{A}'$ from a M-SIS instance and then alter $\mathbf{B}$ in such a way that a forgery results in short vector $\mathbf{x}^*$ such that $\mathbf{A}'\mathbf{x}^* = \mathbf{0} \bmod q$, which readily provides an M-SIS solution. In practice, it is better to use a matrix $\mathbf{A}'$ in normal form, that is, $\mathbf{A}' = [\mathbf{I}|\mathbf{A}]$, as it allows to reduce the size of the public key while retaining security under M-SIS. Plugging an extractable zero-knowledge proof on top of that has no significant effect. Indeed, thanks to the extractability property, the reduction can extract a forged credential from an illicit presentation and then follow the same procedure to derive an M-SIS solution.

As explained earlier, it is useful to decorrelate the signature modulus q and the zero-knowledge proof modulus q_π , but it leads to lifting problems. We indeed recall that the key issue here is that proving that $\mathbf{A}'\mathbf{w} = q\mathbf{j} \bmod q_\pi\mathcal{R}$ does not necessarily mean that $\mathbf{A}'\mathbf{w} = \mathbf{0} \bmod q\mathcal{R}$. In previous works, this was managed by selecting a modulus q_π that is so large that no wrap-around can occur.

In this paper, we show that this requirement is actually not necessary when one aims at extracting an M-SIS solution. Indeed, let us assume that one has extracted from a LNP proof some vector $\mathbf{w}$ such that $[\mathbf{I}|\mathbf{A}]\mathbf{w} = q\mathbf{j} \bmod q_\pi\mathcal{R}$ with $\|\mathbf{w}\|_2 \leq B$ and $\|\mathbf{j}\|_\infty < B_1$. For convenience, we write $\mathbf{w} = [\mathbf{w}_1|\mathbf{w}_2]^T$, where $\mathbf{w}_1 \in \mathcal{R}^d$ is the part of $\mathbf{w}$ that matches $\mathbf{I}$. Let $a < q$ be such that $q_\pi \bmod q = a$ and b be such that $bq_\pi > qB_1 + B_2$, where B_2 is a bound on $[\mathbf{I}|\mathbf{A}]\mathbf{w}$ in $\mathcal{R}$. One can thus write $[\mathbf{I}|\mathbf{A}]\mathbf{w} = q\mathbf{j} + q_\pi\mathbf{h}$ with $\|\mathbf{h}\|_\infty < b$. We thus have $[\mathbf{I}|\mathbf{A}]\mathbf{w} = a\mathbf{h} \bmod q$ and therefore $[\mathbf{I}|\mathbf{A}]\mathbf{w}' = \mathbf{0} \bmod q\mathcal{R}$ where $\mathbf{w}'$ is $[\mathbf{w}_1 - a\mathbf{h}|\mathbf{w}_2]^T$. One can note that $\|\mathbf{w}'\|_2 \leq B + a \cdot b\sqrt{nd}$, which remains small compared to B , provided that one enforces mild requirements on a and b . In other words, we are still able to extract a short witness for the proven relation modulo q if we have controlled (i.e., with b not too large) wrap-arounds modulo q_π . Of course, this approach entails a slack on the extracted witness but we stress that this slack is additive with respect to the M-SIS solution $\mathbf{x}^*$ from above, and can be insignificant in practice by for example choosing a not-too-small proof modulus q_π such that $a = 1$. In all cases, this replaces the initial constraint $q_\pi > qB_1 + B_2$ by $bq_\pi > qB_1 + B_2$, which is significantly milder in practice.

Refined Johnson-Lindenstrauss Bounds. Our second contribution is to revisit the approximate range proof results in [LNP22] with the goal of supporting smaller projection dimensions for $\mathbf{R}$, i.e., the number of rows of $\mathbf{R}$, while also relaxing the constraint on q_π . As mentioned above, there results are based on very technical lemmas for which we propose different proof strategies. This results in several concrete optimizations. First, we manage to replace the original constraint $q_\pi > 41mB_{arp}$ by $q_\pi > C_1\sqrt{m}B_{arp}$ for some constant C_1 , which is significant in practice as the dimension m of the witness is usually quite large.

Second, we reduce the projection dimension to its minimum, that is λ rows for a security level of λ bits, as opposed to 2λ rows in [LNP22]. This reduces the size of the masked opening $\mathbf{z}$ from above, as well as the commitment to the mask $\mathbf{y}$. Alternatively, one can see it as an opportunity to produce two separate approximate range proofs for the same price, which can be interesting when the approximate range proof witness vector $\mathbf{x}$ contains heterogeneous components. Since this contribution is fairly technical, we postpone its presentation to Section 4.3.

Revisiting Garbage Polynomials. Our next contribution focuses on the garbage polynomials $\mathbf{g}$ whose purpose is to mask all the coefficients of $f(\mathbf{w})$ that are unnecessary for the relations we want to prove. More precisely, the approximate range proof also relies on the fact that the masked opening $\mathbf{z}$ is well-formed which involve integer relations that are functions of the witness (and other committed values). These integer relations of well-formedness translate into constraints on the constant coefficient of polynomials. This is also the case of exact norm equations like $\|\mathbf{w}\|_2 = B$, which translate to the constant coefficient of $\mathbf{w}^* \mathbf{w} - B^2$ being 0. The resulting polynomials are aggregated, with random challenge weights, to produce some polynomials $f'_1, \dots, f'_\ell$ that must then have their constant coefficient equal to 0. The entries $\mathbf{g}[i]$ are here to mask all the remaining coefficients of f'_i that may leak information on the witness. To still allow the verifier to check the f'_i indeed have a zero constant coefficient, $\mathbf{g}[i]$ must also have a zero coefficient at this position and be uniformly random coefficients everywhere else¹. As $\mathbf{g}$ serves as a mask, it cannot be revealed to the verifier. Moreover it must not be changed after seeing the challenge so as to retain soundness of the argument. As mentioned above, this has led the authors of [LNP22] to commit to these garbage terms using the ABDLOP scheme, which is rather costly.

If we look a bit closer at the details, this part of the commitment is $\mathbf{t}_g = \mathbf{B}\mathbf{s} + \mathbf{g}$ for some public matrix $\mathbf{B}$ and some short secret vector $\mathbf{s}$. Note that $\mathbf{s}$ is fixed (under the MSIS assumption) by the other parts of the commitment, as proved in [BDL⁺18], hence the binding property despite the large norm of $\mathbf{g}$. If we take a step back, we can identify some source of inefficiency here. $\mathbf{B}\mathbf{s}$ is indeed deterministically generated to mask $\mathbf{g}$ which is itself used as mask in the protocol. Our key observation is that since $\mathbf{B}\mathbf{s}$ is deemed good enough to serve as a mask, why not use it directly, instead of $\mathbf{g}$? Of course, there are a few subtleties to address as we need some zero-coefficients in $\mathbf{g}$ but we show that this can be dealt with using the features of the LNP framework. This also affects the zero-knowledge property of LNP as we can no longer readily rely on the hiding property of the ABDLOP commitment scheme, but we show how to adapt the proof to these specific masks. In the end, it results in a simpler protocol where the commitment $\mathbf{t}_g$ virtually becomes useless. More concretely, we only need to

¹ An optimization of this approach, implemented in our paper, additionally requires to set another coefficient at 0 but we don't consider it here for sake of clarity.

transmit two coefficients per entry of $\mathbf{t}_g$, which is a $32\times$ improvement compared to the previous situation.

A More Efficient Anonymous Credential System. Eventually, we plug all these contributions in the anonymous construction from [JS26], which is itself derived from [AGJ⁺24]. We inherit the security under the M-SIS and M-LWE assumptions from these constructions, which we believe to be a very important feature to be vetted by cybersecurity agencies. In the process, we implement a few optimizations, e.g., related to the user’s secret key, to further improve performance. This however leads us to question the models considered in [BLNS23] and [AGJ⁺24,JS26] (the latter two being based on the one in [FHS19]) which, in our opinion, do not entirely reflect what is expected from an anonymous credentials. In a few words, the main problem stems from defining what a forgery is when the organization does not even know what it has signed. The models cited above address that by either extracting all the signed attributes or by following a “one-more” approach (in the spirit of blind signatures) but this leads to some limitations as we explain in Section 3. In parallel, anonymity is formalized in both cases by the inability of an adversary to distinguish the presentation of a credential cred_0 from the one of a credential cred_1 . There is however a major restriction. If $\{a_i^{(j)}\}_i$ is the set of attributes certified by cred_j (for $j = 0, 1$), then one must have $a_i^{(0)} = a_i^{(1)}$ for all $i \in \mathcal{I}$, where $\mathcal{I}$ is the set of indices of revealed attributes. Although it may look like a natural restriction, this does not capture the fact that the adversary may have obtained no information on these attributes at the issuance time or during previous presentations. In other words, even if the set of revealed attributes differ for these two credentials, the adversary should remain unable to distinguish them unless a previous interaction trivially allows it to do so.

Far from being subordinate, these questions raise very practical issues when it comes to designing and proving the security of anonymous credentials. For example, what is the point of hiding some user’s attributes at the issuance time if the model only requires indistinguishability of credentials with the same subset of revealed credentials? One could for example reveal all the attributes at the issuance time and, during presentation, disclose the necessary attributes and produce a proof of knowledge of the credential and of the concealed attributes. Such a construction would easily achieve the previous anonymity notions and would lead to more efficient constructions as there will be no need for randomness at the issuance time (randomness that is then carried over to the credential, affecting the complexity of all presentations). Similarly, what is the point of binding credentials to the user’s secret key if the model does not allow the adversary to observe presentations by honest users?

To provide a clearer baseline for assessing anonymous credential systems, we therefore present a game-based model that includes the considerations above. We believe it is necessary to better understand, and formalize, the differences with previous works. Our model can then be seen as an extension of the models used in prior constructions like [BLNS23,AGJ⁺24], supporting the blind issuance of

credentials and user secret keys for device-binding. Nevertheless, we stress that we do not claim comprehensiveness for it as we, for example, do not consider situations where one proves some predicate about a hidden attribute, e.g., age proofs.

Performance. As our model is more permissive and gives more freedom to the adversary, it imposes further constraints on the design and parameters to reach the same security level. Nevertheless, even with this performance overhead, we demonstrate significant improvements over the previous construction based on standard assumptions [JS26], with credential presentation proof sizes of 35.45 KB instead of 46.89 KB in [JS26] which represents a 25% gain.

Our performance then approaches the one of anonymous credentials based on ad-hoc interactive assumptions [BLNS23], without needing the latter, which we believe is a primary aspect for practical deployment. We compare our scheme with existing anonymous credentials in Section 7, both in terms of performance and security assurances provided by the security models, proofs and assumptions.

2 Preliminaries

For integers $m \leq n$, we define $[m, n] = \{m, \dots, n\}$, and $[n] = [1, n]$. The unit group of a ring R is denoted by $R^\times$, and we abuse the notation by defining $\mathbb{N}^\times = \mathbb{N} \setminus \{0\}$. We use bold font for vectors $\mathbf{a}$ (lower case) and matrices $\mathbf{A}$ (upper case). The transpose is then denoted by a superscript as $\mathbf{A}^T$ or $\mathbf{a}^T$. We denote by $\mathbf{a}[j]$ the j -th entry of a vector $\mathbf{a}$. Further, for $q \in \mathbb{N}^\times$, we let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ which can be identified by the set of representatives $[0, q-1]$ or $(-q/2, q/2] \cap \mathbb{Z}$. When the centered representation of some $a \in \mathbb{Z}_q$ is needed, we use $a \bmod^\pm q$. Finally, $\|\mathbf{a}\|_p$ denotes the standard ℓ^p norm of $\mathbf{a} \in \mathbb{R}^n$, and $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$ the spectral norm of matrix $\mathbf{A}$.

2.1 Algebraic Number Theory

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

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

$\tau^{-1}([a_0| - a_{n-1}| \dots | - a_1]^T)$. These notations naturally extend to other the base rings $\mathcal{R}_q$ or $\mathcal{K}_{\mathbb{R}}$, and also to vectors and matrices entry-wise, except that the conjugate of a matrix actually corresponds to the conjugate transpose. For an integer η , we sometimes use the sets $T_\eta = \tau^{-1}([0, \eta]^n)$ and $S_\eta = \tau^{-1}([-\eta, \eta]^n)$. We also define the usual vector norms $\|\cdot\|_p$ over $\mathcal{R}$ by $\|r\|_p := \|\tau(r)\|_p$, and the spectral norm $\|\mathbf{A}\|_2$ by $\|M_\tau(\mathbf{A})\|_2$.

2.2 Lattices

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

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 . For $\eta \in \mathbb{N}^\times$, we define the centered binomial distribution ψ_η as the distribution of $\sum_{i \in [\eta]} a_i - b_i$ for $a_1, b_1, \dots, a_\eta, b_\eta$ independently drawn from $U(\{0, 1\})$. The distribution ψ_η is supported over $[-\eta, \eta]$ and has mass $\psi_\eta(k) = 2^{-2\eta} \binom{2\eta}{\eta+k}$ for all $k \in [-\eta, \eta]$. We then use $\mathcal{B}_\eta$ to denote the distribution over $\mathcal{R}$ whose coefficients follow ψ_η , i.e., $\tau^{-1}(\psi_\eta^n)$. We use $x \leftarrow \mathcal{P}$ to note the action of sampling $x \in S$ according to the probability distribution $\mathcal{P}$. In contrast, we use $x \sim \mathcal{P}$ when the random variable x follows $\mathcal{P}$.

Discrete Gaussian Distributions. For a lattice (coset) $A \subseteq \mathbb{R}^d$, we define the *discrete Gaussian distribution* $\mathcal{D}_{A, \sqrt{\mathbf{S}}, \mathbf{c}}$ of support A , positive-definite covariance $\mathbf{S} \in \mathbb{R}^{d \times d}$ and center $\mathbf{c} \in \mathbb{R}^d$ by its density $\mathcal{D}_{A, \sqrt{\mathbf{S}}, \mathbf{c}} : \mathbf{x} \in A \mapsto \rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathbf{x}) / \sum_{\mathbf{y} \in A} \rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathbf{y})$, where $\rho_{\sqrt{\mathbf{S}}, \mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x} - \mathbf{c})^T \mathbf{S}^{-1}(\mathbf{x} - \mathbf{c}))$. 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 module lattice cosets $M \subseteq \mathcal{K}_{\mathbb{R}}^d$ with $\mathbf{c} \in \mathcal{K}_{\mathbb{R}}^d$ and $\mathbf{S} \in \mathbb{R}^{nd \times nd}$, we extend the notation to $\mathcal{D}_{M, \sqrt{\mathbf{S}}, \mathbf{c}} := \tau^{-1}(\mathcal{D}_{\tau(M), \sqrt{\mathbf{S}}, \tau(\mathbf{c})})$. When $M = \mathcal{R}^d$, $\tau(M) = \mathbb{Z}^{nd}$ which means the distribution corresponds to sampling an integer vector according to $\mathcal{D}_{\mathbb{Z}^{nd}, \sqrt{\mathbf{S}}, \tau(\mathbf{c})}$ and defining a vector of $\mathcal{R}^d$ via τ^{-1} . As coined by Micciancio and Regev [MR07], we define the *smoothing parameter* of a lattice $\mathcal{L}$, parameterized by $\varepsilon > 0$, by $\eta_\varepsilon(\mathcal{L}) = \inf\{s > 0 : \rho_{1/s}(\mathcal{L}^*) = 1 + \varepsilon\}$. We first need a Gaussian regularity lemma from [GPV08, Lem. 5.2] generalized to non-spherical distributions. We state it over rings for coherence but it also applies to the integers.

Lemma 2.1 ([GPV08, Lem. 5.2] adapted). *Let d, m, q be positive integers, and $\overline{\mathbf{A}} \in \mathcal{R}_q^{d \times m}$ such that $\overline{\mathbf{A}}\mathcal{R}_q^m = \mathcal{R}_q^d$. 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 $\mathcal{P} = \overline{\mathbf{A}}\mathcal{D}_{\mathcal{R}^m, \sqrt{\mathbf{S}}} \bmod q\mathcal{R}$. It holds that $\forall \mathbf{x} \in \mathcal{R}_q^d, \mathcal{P}(\mathbf{x}) \in [(1 - \varepsilon)/(1 + \varepsilon), 1 + \varepsilon]q^{-nd}$.*

We now give the standard discrete Gaussian tail bounds from [Ban93,Ban95].

Lemma 2.2 ([Ban93, Lem. 1.5] & [Ban95, Lem. 2.10]). *Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d , and $s > 0$. Then, for all $c > 1/\sqrt{2\pi}$ and $t > 0$, we have*

$$\begin{aligned}\mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L},s}} \left[\|\mathbf{x}\|_2 > cs\sqrt{d} \right] &< \left(c\sqrt{2\pi}ee^{-\pi c^2} \right)^d \\ \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\mathcal{L},s}} \left[\|\mathbf{x}\|_\infty > ts \right] &< 2de^{-\pi t^2}.\end{aligned}$$

Unless the tailcut probability is specified, we target a probability of at most $2^{-(\lambda+O(1))}$, and let c_d be the smallest $c > 1/\sqrt{2\pi}$ such that $(c\sqrt{2\pi}ee^{-\pi c^2})^d \leq 2^{-(\lambda+O(1))}$, where λ is the implicit security parameter.

Gadget Sampler. Our construction leverages the preimage sampler **TSampler** of [JS26]. For a matrix $\mathbf{A}_t = [\mathbf{I}_d | \mathbf{A} | q_L t \mathbf{G}_H - (\mathbf{R}_1 + \mathbf{A} \mathbf{R}_2) | \mathbf{a}] \bmod q\mathcal{R}$, with $q = q_L b_H^{k_H}$, $\mathbf{A} \in \mathcal{R}_q^{d \times d}$, $\mathbf{G}_H = [1|b_H| \dots |b_H^{k_H-1}] \otimes \mathbf{I}_d$, $\mathbf{a} \in \mathcal{R}_q^d$, $\mathbf{R}_1, \mathbf{R}_2$ in $\mathcal{R}^{d \times dk_H}$ with small spectral norms, and $t \in \mathcal{R}_{b_H}^\times$ a tag, it samples, for any $\mathbf{u} \in \mathcal{R}_q^d$, a Gaussian $\mathbf{v} \in \mathcal{R}^{2d+dk_H+1}$ such that $\mathbf{A}_t \mathbf{v} = \mathbf{u} \bmod q\mathcal{R}$. More precisely, the authors of [JS26] prove $\mathbf{v}$ follows a Gaussian statistically close to $\mathcal{D}_{\mathcal{L}_q^u(\mathbf{A}_t), \sqrt{S}}$ for $\mathbf{S} = \text{diag}(s_1^2 \mathbf{I}_{nd}, s_2^2 \mathbf{I}_{nd}, s_3^2 \mathbf{I}_{n(dk_H+1)})$ with

$$s_1^2 = \frac{s_L^4}{s_L^2 - \eta^2} \cdot \frac{b_L^{2k_L} - 1}{b_L^2 - 1} + 3 \frac{s_H^4}{s_H^2 - \eta^2} \|\mathbf{R}_1\|_2^2, \quad s_2^2 = 3 \frac{s_H^4}{s_H^2 - \eta^2} \|\mathbf{R}_2\|_2^2, \quad s_3^2 = 3 \frac{s_H^4}{s_H^2 - \eta^2},$$

with $q_L = b_L^{k_L}$, $s_X = b_X \eta_{\varepsilon/dk_X}(\mathbb{Z}^n)$ and $\eta = \eta_\varepsilon(\mathbb{Z}^{nd(k_L+k_H)})$. The factors 3 can actually be tweaked with slack parameters γ_2, γ_3 [JS26, Cor. 3.1]. We refer to [JS26] for more details on the sampler and its security.

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

Lemma 2.3 ([LNS21, Lem. 3.2]). *Let d be a positive integer, and $S \subset \mathbb{R}^d$ a set of vectors of Euclidean norm at most $T > 0$. Let $\mathcal{D}_S$ be a distribution over S . Let $M > 1$, $\alpha = \sqrt{\pi/\ln M}$ and $s \geq \alpha T$. We then define the following distributions.*

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

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

2.4 Hardness Assumptions

The security of our construction is based on the *Module Learning With Errors* (M-LWE) and *Module Short Integer Solution* (M-SIS) problems [LS15]. For a problem P where the advantage of an adversary $\mathcal{A}$ is called $\text{Adv}_P[\mathcal{A}]$, we define the hardness bound of P by $\varepsilon_P = \sup_{\mathcal{A}} \text{ppT} \text{Adv}_P[\mathcal{A}]$, where the parameters of the problem are clear from the context.

Definition 2.1 (M-SIS). Let $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ 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 $\text{M-SIS}_{n,d,m,q,\beta}$ 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 $\mathbf{A} \leftarrow U(\mathcal{R}_q^{d \times m-d})$. The advantage of an adversary $\mathcal{A}$ is $\text{Adv}_{\text{M-SIS}}[\mathcal{A}] = \mathbb{P}_{\mathbf{x} \leftarrow \mathcal{A}(\mathbf{A})} [\mathbf{x} \in \mathcal{L}_q^\perp([\mathbf{I}_d | \mathbf{A}']) \wedge 0 < \|\mathbf{x}\|_2 \leq \beta]$.

Definition 2.2 (M-LWE). Let $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ with n a power-of-two. Let d, m, q be positive integers and $\mathcal{D}$ be a distribution over $\mathcal{R}^{d+m}$. The Module Learning With Errors problem, denoted $\text{M-LWE}_{n,d,m,p,\mathcal{D}}$ asks to distinguish between $\mathcal{P}_0 = U(\mathcal{R}_q^{m \times d} \times \mathcal{R}_q^m)$ and the distribution $\mathcal{P}_1$ of $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q\mathcal{R})$, where $\mathbf{A} \sim U(\mathcal{R}_q^{m \times d})$ and $\mathbf{x} = [\mathbf{s}^T | \mathbf{e}^T]^T \sim \mathcal{D}$. We define $\text{Adv}_{\text{M-LWE}}[\mathcal{A}] = |\mathbb{P}_{(\mathbf{A}, \mathbf{b}) \sim \mathcal{P}_0}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1] - \mathbb{P}_{(\mathbf{A}, \mathbf{b}) \sim \mathcal{P}_1}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = 1]|$ as the advantage of a PPT adversary in solving M-LWE.

3 Security Models of Anonymous Credentials

Informally, anonymous credentials refer to a system involving three types of entities, the organization (*a.k.a* the issuer), the users and the verifiers, where users can request credentials from the organization on a set of attributes, some of which may be hidden, and then present these credentials to verifiers to attest the validity of some of their attributes. The key difference compared to standard digital signatures lies in the privacy assurances for the users. The latter are indeed ensured that no information will leak beyond what they agree to reveal, which implies unlinkability and secrecy of undisclosed attributes.

Unfortunately the broad consensus on these general guidelines does not extend to formal security definitions that are quite difficult to identify for several reasons. One of these reasons is the parentage from standard digital signatures that incites to define a similar unforgeability notion. In particular, intuitively, it is tempting to see an adversary that would make use of a forged certificate as a valid attacker. However, to decide if a certificate is forged, one needs to know which sets of attributes were signed during the issuance procedures. This can be done by requiring that all the attributes are known to the issuer, as in [FHS19] but one then loses a very important feature of anonymous credentials, namely blind issuance. A more subtle approach, as in [LCL⁺23], is to allow issuance on concealed attributes in the real world but to define a model where issuance oracles can only be queried on fully disclosed attributes. The discrepancy between the real world and the model is then managed by requiring from the users, in the issuance process, a straight-line extractable proof of knowledge of all the attributes. This way, the reduction can convert real-world queries into model queries by extracting all the attributes. However, this comes at the cost of significantly increasing the complexity of the issuance proof. Beyond that, this model prescribes some components for the anonymous credentials systems, which is not ideal. On this unforgeability aspect, we therefore prefer to follow the model in [BLNS23] that we extend to cover some attacks that were not considered, as we will explain.

Another challenge when defining a formal security model is the user secret key usk considered by some constructions. Concretely, the underlying question is: should we bind the issued credential to a user’s secret key? Nowadays, this question is often related to the notion of device-binding which requires that a credential shall not be used without the support of a secure element that would store the usk . However, the binding to a secret key serves a more general purpose, which is to ensure that only the users can present the certificates they receive. At first sight, this may seem unnecessary, in particular in cases such as [BLNS23] where the presentation of the credential essentially consists in a zero-knowledge proof of knowledge of the said credential. This way, the credential remains perfectly hidden from the adversary that will never be able to use it. Unfortunately, this would, here again, essentially prescribe some building blocks for the presentation and thus make the model less general than it should be. In particular, it would rule out constructions of anonymous credentials based on re-randomizable signatures (e.g. [CL04, PS16]) where the (randomized) certificates are sent in clear. If these certificates were not bound to a user’s secret key, then any verifier (or adversary with access to the presentation data) would be able to use these certificates, provided that it knows the associated attributes. The latter condition may open up for an alternative to the user secret key that was, for example, considered in [BLNS23]. Concretely, instead of formally defining a user secret key, one could prevent the previous attack by requiring that at least one attribute (with enough entropy) remains undisclosed. In other words, this attribute plays the role of usk . However, this is not optimal, as we will illustrate with our construction. Intuitively, the key issue is that an attribute and the user’s secret key face different requirements: while an attribute must be immutable, the user’s secret key may be altered in some ways as long as it remains impossible for an adversary that does not know it to use the associated certificates.

3.1 Syntax

We consider a system where the organization may issue credentials on a vector $\mathbf{m}$ of m attributes. For each $i \in [m]$, we denote by S_i the i -th attribute space that is clear from the context (e.g., the set of cities of a given countries, the set of all possible birthdates up to some reasonable limit, etc.). We also define the space for attribute vectors as $S = \times_{i \in [m]} S_i$. Said differently, an attribute vector $\mathbf{m}$ should always lie in S , and by extension each attribute $\mathbf{m}[i]$ belongs to S_i . For $\mathcal{I} \subseteq [m]$, we define $\mathbf{m}(\mathcal{I})$ as the vector of $\times_{i \in [m]} (S_i \cup \{\emptyset\})$ such that $\mathbf{m}(\mathcal{I})[i] = \mathbf{m}[i]$ if $i \in \mathcal{I}$, and $\mathbf{m}(\mathcal{I})[i] = \emptyset$ if $i \in [m] \setminus \mathcal{I}$. It can be seen as the vector $\mathbf{m}$ whose entries have been redacted for indices outside of $\mathcal{I}$.

The credential issued on the attribute vector $\mathbf{m}$ fixes not only the values of the attributes but also their position within $\mathbf{m}$. In particular, an adversary managing to derive from a credential cred on $\mathbf{m}$ a credential cred' on a permutation of $\mathbf{m}$ is considered successful. Reflecting our discussion above, we do consider a user key pair that is not a mere attribute. When requesting or presenting a credential, the user can define a vector of disclosed attributes $\mathbf{m}(\mathcal{I})$ where $\mathcal{I}$ is some subset of $[m]$. We define $\bar{\mathcal{I}} = [m] \setminus \mathcal{I}$. Note that the user is free to select different sets $\mathcal{I}$ for

each issuance or presentation. In particular, selecting some set $\mathcal{I}$ when requesting a credential does not preclude any option when presenting this credential.

Finally, we note that our model does not consider the case where one would claim some statement about a hidden attribute (e.g., $\mathbf{m}[i] \geq x$). This is a deliberate choice as this feature dramatically increases the complexity of the model, in particular in the anonymity experiment where the definition of a valid challenge becomes significantly more difficult to formalize. We nevertheless stress that this feature can be readily added to our construction as we do not hash our attributes, contrarily to, e.g., [BLNS23].

Definition 3.1. *An anonymous credentials system is defined as a set of algorithms and protocols ([Setup](#), [OKeyGen](#), [UKeyGen](#), [Issue](#), [Show](#)) where*

- [Setup](#)(1^λ). *On input the security parameter λ , it generates the public parameters $\mathbf{pp}$ (which include λ);*
- [OKeyGen](#)($\mathbf{pp}$). *On input $\mathbf{pp}$, it generates the organization’s key pair $(\mathbf{opk}, \mathbf{osk})$;*
- [UKeyGen](#)($\mathbf{pp}$). *On input $\mathbf{pp}$, it generates the user’s key pair $(\mathbf{upk}, \mathbf{usk})$;*
- [Issue](#)($O(\mathbf{osk}, \mathbf{opk}, \mathbf{upk}, \mathbf{pp}, \mathbf{m}(\mathcal{I})), U(\mathbf{usk}, \mathbf{upk}, \mathbf{opk}, \mathbf{pp}, \mathbf{m}, \mathcal{I})$). *It is a protocol between an organization O , holding $(\mathbf{osk}, \mathbf{opk}, \mathbf{upk}, \mathbf{pp})$ and the user’s disclosed attributes $\mathbf{m}(\mathcal{I})$, and a user U holding $(\mathbf{usk}, \mathbf{upk}, \mathbf{opk}, \mathbf{pp})$ and their attributes $\mathbf{m}$ (with the index set $\mathcal{I}$ corresponding to disclosed attributes). If the protocol did not abort, the organization O gets the credential $\mathbf{cred}$ it produced for U , while the user U gets a possibly unblinded version of the credential we call $\mathbf{cred}_{\text{unb}}$ on its attributes. In case of aborts, each party gets $\perp$;*
- [Show](#)($U(\mathbf{usk}, \mathbf{upk}, \mathbf{opk}, \mathbf{pp}, \mathbf{m}, \mathcal{I}, \mathbf{cred}_{\text{unb}}), V(\mathbf{opk}, \mathbf{pp}, \mathbf{m}(\mathcal{I}))$). *It is a protocol between a user U holding $(\mathbf{usk}, \mathbf{upk}, \mathbf{opk}, \mathbf{pp}, \mathbf{m}, \mathcal{I}, \mathbf{cred}_{\text{unb}})$ and a verifier V holding $(\mathbf{opk}, \mathbf{pp}, \mathbf{m}(\mathcal{I}))$ where $\mathbf{m}(\mathcal{I})$ are the attributes disclosed by the user. The protocol outputs 1 to V if the unblinded credential $\mathbf{cred}_{\text{unb}}$ is valid with respect to the disclosed attributes, and 0 otherwise.*

As opposed to previous models like [FHS19, BLNS23] where the [Issue](#) protocol only specifies the output for the user, we allow the issuer to get an output in case of a successful execution. The output $\mathbf{cred}$ essentially corresponds to what the issuer generates, while $\mathbf{cred}_{\text{unb}}$ may differ from it based on the user randomness in the protocol. Specifying the notations of what is known by the issuer and user at the result of a successful issuance helps defining the desired security properties we expect from anonymous credentials.

3.2 Security Model

Security Notations. Our security games will make use of the following variables and oracles. We start by defining the following variables.

HU: Set, initially empty, of honest user indices.

CU: Set, initially empty, of corrupt user indices.

ctr: Issuance counter, initially 0.

ctr_{cor}: number of certificates owned by corrupt users, initially 0.

State_C: Set, initially empty, tracking the knowledge of the challenger $\mathcal{C}$ in the experiments. The set is populated after each issuance query (via $\mathcal{O}_{\text{ObtIss}}$, $\mathcal{O}_{\text{Obtain}}$, or $\mathcal{O}_{\text{Issue}}$) by tuples of the form $(j, l, x_l, \text{attr}_l)$, where j is a user index, l an issuance index, x_l the (possibly unblinded) credential produced in the l -th issuance query, and attr_l is a vector containing the attributes that are certified by x_l and known to $\mathcal{C}$. Said differently, $\text{attr}_l \in \times_{i \in [m]} (S_i \cup \{\emptyset\})$. If $\text{attr}_l[i] = \emptyset$, it means $\text{attr}_l[i]$ is not currently known to the challenger $\mathcal{C}$.

State_A: Set, initially empty, tracking the knowledge of the adversary $\mathcal{A}$ in the experiments. In particular, we assume **State_A** is known to $\mathcal{A}$ at any stage of the games. The set is populated after each issuance query (via $\mathcal{O}_{\text{ObtIss}}$, $\mathcal{O}_{\text{Obtain}}$, or $\mathcal{O}_{\text{Issue}}$) or show query ($\mathcal{O}_{\text{Show}}$) by tuples of the form $(j, l, x_l, \text{attr}_l)$, where j is a user index, l an issuance index, x_l the (possibly unblinded) credential produced in the l -th issuance query, and attr_l is a vector containing the attributes that are certified by x_l and known to $\mathcal{A}$. Said differently, $\text{attr}_l \in \times_{i \in [m]} (S_i \cup \{\emptyset\})$. If $\text{attr}_l[i] = \emptyset$, it means $\text{attr}_l[i]$ is not currently known to the adversary $\mathcal{A}$.

We then define the following oracles which are used in either the anonymity game or the unforgeability game, or both. These oracles potentially update the above variables throughout the experiments.

- $\mathcal{O}_{\text{HU}}(j)$** : It registers user j as an honest user, if it does not already exist. Given a user index j , it returns $\perp$ if $j \in \text{HU} \cup \text{CU}$. Otherwise, it performs $(\text{upk}_j, \text{usk}_j) = \text{UKeyGen}(\text{pp})$, updates $\text{HU} = \text{HU} \cup \{j\}$ and returns upk_j .
- $\mathcal{O}_{\text{CU}}(j, \text{upk})$** : It corrupts user j , if it was honest, or registers user j as a corrupt user. Given a user index j and (optionally) a public key upk , it registers a new corrupt user with public key upk if $j \notin \text{HU}$. This models the ability to register users with malformed keys, i.e., who do not necessarily know the associated secret key. Otherwise, if $j \in \text{HU}$, it updates $\text{HU} = \text{HU} \setminus \{j\}$ and $\text{CU} = \text{CU} \cup \{j\}$, and returns usk_j along with all the data in **State_C** corresponding to user j , that is all the tuples $(j, *, *, *) \in \text{State}_C$. Let k be the number of such tuples, it then updates the counter $\text{ctr}_{\text{cor}} = \text{ctr}_{\text{cor}} + k$.
- $\mathcal{O}_{\text{ObtIss}}(j, \mathbf{m})$** : It generates a credential for an honest user j on the given attribute vector $\mathbf{m}$. Given an honest user index $j \in \text{HU}$ and attribute vector $\mathbf{m}$, it runs $(\text{cred}, \text{cred}_{\text{unb}}) = \text{Issue}(O(\text{osk}, \text{opk}, \text{upk}_j, \text{pp}, \mathbf{m}), U_j(\text{usk}_j, \text{upk}_j, \text{opk}, \text{pp}, \mathbf{m}, [m]))$ assuming the roles of both O and user j . If successful, it increments ctr , and updates $\text{State}_C = \text{State}_C \cup \{(j, \text{ctr}, \text{cred}_{\text{unb}}, \mathbf{m})\}$ and $\text{State}_A = \text{State}_A \cup \{(j, \text{ctr}, \text{cred}_{\text{unb}}, \mathbf{m})\}$. We assume here the adversary may learn the unblinded credential emitted to the honest user. This imposes the use of secret user keys to ensure the adversary cannot show this credential. Note that the entire attribute vector $\mathbf{m}$ is known in **State_C** and **State_A**. It then returns $\top$ if the execution succeeded. If $j \notin \text{HU}$, it simply returns $\perp$.
- $\mathcal{O}_{\text{Obtain}}(j, \mathbf{m}(\mathcal{I}))$** : It generates a (possibly maliciously formed) credential to an honest user j on the set of disclosed attributes. Given a user index j and a vector of disclosed attributes $\mathbf{m}(\mathcal{I})$, this oracles returns $\perp$ if $j \notin \text{HU}$. Otherwise, it first completes $\mathbf{m}(\mathcal{I})$ into $\mathbf{m}$ with random elements, i.e., for $i \in$

$\bar{\mathcal{I}}, \mathbf{m}[i] \leftarrow U(S_i)^2$ and $\mathbf{m}[i] = \mathbf{m}(\mathcal{I})[i]$ for $i \in \mathcal{I}$. It then runs $(\text{cred}, \text{cred}_{\text{unb}}) = \text{Issue}(\mathcal{A}(\cdot), U(\text{usk}_j, \text{upk}_j, \text{opk}, \text{pp}, \mathbf{m}, \mathcal{I}))$ with the adversary $\mathcal{A}$ posing as the organization. If successful, it increments ctr , and updates $\text{State}_{\mathcal{C}} = \text{State}_{\mathcal{C}} \cup \{(j, \text{ctr}, \text{cred}_{\text{unb}}, \mathbf{m})\}$ and $\text{State}_{\mathcal{A}} = \text{State}_{\mathcal{A}} \cup \{(j, \text{ctr}, \text{cred}, \mathbf{m}(\mathcal{I}))\}$. Here, $\mathcal{A}$ only gets the redacted attribute vector $\mathbf{m}(\mathcal{I})$.

$\mathcal{O}_{\text{Issue}}(j, \mathbf{m}(\mathcal{I}))$: It generates a credential to a corrupt user j on the given disclosed attributes. Given a user index j and disclosed attributes $\mathbf{m}(\mathcal{I})$, it returns $\perp$ if $j \notin \text{CU}$. Else, it runs $(\text{cred}, \text{cred}_{\text{unb}}) = \text{Issue}(O(\text{osk}, \text{opk}, \text{upk}_j, \text{pp}, \mathbf{m}(\mathcal{I})), \mathcal{A}(\cdot))$ where the adversary plays the corrupt user. If successful, it increments ctr and ctr_{cor} , and updates $\text{State}_{\mathcal{C}} = \text{State}_{\mathcal{C}} \cup \{(j, \text{ctr}, \text{cred}, \mathbf{m}(\mathcal{I}))\}$ and $\text{State}_{\mathcal{A}} = \text{State}_{\mathcal{A}} \cup \{(j, \text{ctr}, \text{cred}_{\text{unb}}, \mathbf{m}(\mathcal{I}))\}$. Here, both $\mathcal{C}$ and $\mathcal{A}$ only get the redacted attribute vector $\mathbf{m}(\mathcal{I})$. As $\mathcal{A}$ plays the user, it potentially knows other attributes, but it could also run the protocol dishonestly without having other attributes.

$\mathcal{O}_{\text{Show}}(l, \mathcal{I})$: It shows the l -th emitted credential for a given set of indices of disclosed attributes. Given an issuance index l and a set $\mathcal{I}$ of indices of disclosed attributes, it returns $\perp$ if $l > \text{ctr}$ (meaning there is currently no issuance with index l). If $l \leq \text{ctr}$, the oracle retrieves the corresponding entry $(j, l, x_l, \text{attr}_l)$ from $\text{State}_{\mathcal{C}}$. It aborts if $j \notin \text{HU}$. Else, it means x_l has necessarily been produced through $\mathcal{O}_{\text{ObtLss}}$ or $\mathcal{O}_{\text{Obtain}}$, meaning attr_l has no redacted entries, and thus $\mathbf{m} = \text{attr}_l \in S$. It also means x_l is an unblinded credential $\text{cred}_{\text{unb}, l}$, which can therefore be shown. It can then run $(\emptyset, b) = \text{Show}(U(\text{usk}_j, \text{upk}_j, \text{opk}, \text{pp}, \mathbf{m}, \mathcal{I}, \text{cred}_{\text{unb}, l}), \mathcal{A}(\cdot))$ with the adversary posing as the verifier. It then updates the set of known attributes for $\mathcal{A}$: it recovers $(j, l, x_l^*, \text{attr}_l^*)$ in $\text{State}_{\mathcal{A}}$, and sets $\text{attr}_l^*[i] = \mathbf{m}[i]$ for all $i \in \mathcal{I}$. Notice that this set $\mathcal{I}$ possibly differs from that of the issuance, in which case the adversary can gradually learn more attributes from queries to $\mathcal{O}_{\text{Show}}$.

Anonymity. In formalizing the anonymity and unforgeability, we use the sets $\text{State}_{\mathcal{C}}$ and $\text{State}_{\mathcal{A}}$ to define the least possible restrictions on the adversary success conditions. In the anonymity game, keeping track of what the adversary knows about the honest users' credential allows for significantly alleviating the restrictions on the challenge certificates. In particular, we stress that the attributes corresponding to these certificates may differ, even on the set of disclosed attributes (something that is forbidden in usual anonymity experiments, e.g., [BLNS23]), as long as the adversary has not asked to reveal these different attributes in an oracle query (either during issuance, or subsequent showings).

Definition 3.2. *An anonymous credentials system is anonymous if, for any PPT adversary $\mathcal{A}$, $\text{Adv}_{\text{an}}[\mathcal{A}] = |\mathbb{P}[\text{Game}_{\text{an-0}}(\mathcal{A}) = 1] - \mathbb{P}[\text{Game}_{\text{an-1}}(\mathcal{A}) = 1]|$ is negligible in λ , where $\text{Game}_{\text{an-b}}(\mathcal{A})$ is defined in Figure 3.1 for $b \in \{0, 1\}$.*

Unforgeability. As discussed above, defining unforgeability in a context where the signed attributes may be unknown is quite difficult. Our definition uses

² The completed attributes may be chosen arbitrarily rather than uniformly.

```

Gamean-b( $\mathcal{A}$ ):
1  $\text{HU}, \text{CU}, \text{State}_{\mathcal{C}}, \text{State}_{\mathcal{A}} \leftarrow \emptyset$ , and  $\text{ctr} \leftarrow 0$ 
2  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ 
3  $(\text{osk}, \text{opk}) \leftarrow \text{OKeyGen}(\text{pp})$ 
4  $(l_0, l_1, \mathcal{I}) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{Obtain}}, \mathcal{O}_{\text{Show}}}(\text{opk}, \text{osk})$ 
5 if  $\llbracket \min(l_0, l_1) > \text{ctr} \rrbracket$  abort
6 Look up  $\text{State}_{\mathcal{C}}$  for  $(j_0, l_0, \text{cred}_{\text{unb}, l_0}, \text{attr}_{l_0})$  and  $(j_1, l_1, \text{cred}_{\text{unb}, l_1}, \text{attr}_{l_1})$ 
7 Execute  $(\emptyset, 1) \leftarrow \text{Show}(\mathcal{C}(\text{usk}_{j_b}, \text{opk}, \text{pp}, \text{attr}_{l_b}, \mathcal{I}, \text{cred}_{\text{unb}, l_b}), \mathcal{A}(\cdot))$ 
8  $b^* \leftarrow \mathcal{A}^{\mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{Obtain}}, \mathcal{O}_{\text{Show}}}(\text{opk}, \text{osk})$ 
9 Look up  $\text{State}_{\mathcal{A}}$  for  $(j_0, l_0, x_{l_0}, \text{attr}_{l_0}^*)$  and  $(j_1, l_1, x_{l_1}, \text{attr}_{l_1}^*)$ 
10 if  $\llbracket j_0 \in \text{CU} \rrbracket \vee \llbracket j_1 \in \text{CU} \rrbracket$  abort
11 for each  $i \in \mathcal{I}$ , abort if none of the following conditions hold:
    (1)  $\text{attr}_{l_0}^*[i] = \emptyset = \text{attr}_{l_1}^*[i]$     (2)  $\text{attr}_{l_0}[i] = \text{attr}_{l_1}[i]$ 
12 return  $b^*$ 

```

Fig. 3.1. Anonymity experiment

the “one-more” unforgeability experiment of [BLNS23] as a baseline. It requires that no adversary can make more than ctr_{cor} inconsistent presentations if it has received ctr_{cor} certificates. By inconsistent presentations, we mean presentations that would for example disclose an attribute $\mathbf{m}[i]$ the first time and another one $\mathbf{m}'[i] \neq \mathbf{m}[i]$ the second time. This kind of success conditions, reminiscent of blind signatures, avoids extraction of the full list of attributes each time the adversary queries an issuance oracle, which in turn loosens constraints on the design of the issuance process.

We nevertheless note that this does not cover all attack scenarios. In particular, let us assume that an adversary has made two issuance queries where $\mathbf{m}[i]$ was systematically disclosed for some i and then presents a credential with a disclosed attribute $\mathbf{m}^*[i]$ that differs from all the previous $\mathbf{m}[i]$. This is clearly an attack as no credential should have allowed that. And yet this is not seen as such in the previous model since there is only one presentation for two issuances.

To address that, we augment the success condition to cover both cases: an adversary wins if it can make more than ctr_{cor} inconsistent presentations, as in [BLNS23] or if it can make a presentation that is inconsistent with all the issuances so far, as in [FHS19]. We additionally add a user key and allow the adversary to win by proving a statement that only honest users’ certificates can support. This models the fact that a credential is useless without the associated user key.

Dealing with these different success conditions inherently complexifies the experiment. To retain a simple presentation, we introduce the following notations. First, we define $\text{State}_{\mathcal{S}}$ as the list of attributes $\mathbf{m}(\mathcal{I})$ disclosed by the adversary when running **Show**. Second, we introduce the notation $\neg_X$, for $X \in \{\mathcal{C}, \mathcal{S}\}$, which informally means that an element is inconsistent with all the entries in State_X . More concretely:

- $\mathbf{m}(\mathcal{I}) \neg_{\mathcal{C}} \text{State}_{\mathcal{C}}$ is defined as “ $\forall (j, l, x_l, \text{attr}_l) \in \text{State}_{\mathcal{C}}$ with $j \in \text{CU}$, $\exists i_l \in \mathcal{I}, \emptyset \neq \text{attr}_l[i_l] \neq \mathbf{m}(\mathcal{I})[i_l]$ ”.

- $\mathbf{m}(\mathcal{I}) \dashv_S \text{State}_S$ is defined as “ $\forall \mathbf{m}'(\mathcal{I}') \in \text{State}_S, \exists i \in \mathcal{I} \cap \mathcal{I}', \mathbf{m}(\mathcal{I})[i] \neq \mathbf{m}'(\mathcal{I}')[i]$ ”.

In both cases, it means that we are certain that no element in State_X can support this list of presented attributes. Finally, our experiment involves an additional counter, $\text{ctr}_{\text{inc-show}}$, which represents the number of inconsistent Show.

Definition 3.3. *An anonymous credentials system is unforgeable if, for any PPT adversary $\mathcal{A}$, $\text{Adv}_{uf}[\mathcal{A}] = \mathbb{P}[\text{Game}_{uf}(\mathcal{A}) = 1]$ is negligible in λ , where $\text{Game}_{uf}(\mathcal{A})$ is defined in Figure 3.2.*

```

Gameuf( $\mathcal{A}$ ):
1   $\text{HU}, \text{CU}, \text{State}_C, \text{State}_A, \text{State}_S \leftarrow \emptyset$  and  $\text{ctr}, \text{ctr}_{\text{cor}}, \text{ctr}_{\text{inc-show}} \leftarrow 0$ 
2   $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ 
3   $(\text{osk}, \text{opk}) \leftarrow \text{OKeyGen}(\text{pp})$ 
4  while  $\text{ctr}_{\text{inc-show}} \leq \text{ctr}_{\text{cor}}$ 
5     $\mathbf{m}(\mathcal{I}) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{ObtIss}}, \mathcal{O}_{\text{Issue}}, \mathcal{O}_{\text{Show}}}(\text{opk})$ 
6     $(\emptyset, b) \leftarrow \text{Show}(\mathcal{A}(\cdot), \mathcal{C}(\text{opk}, \text{pp}, \mathbf{m}(\mathcal{I})))$ 
7    if  $\llbracket b = 1 \rrbracket$ :
8      if  $\llbracket \mathbf{m}(\mathcal{I}) \dashv_C \text{State}_C \rrbracket$  then
9        return 1
10     if  $\llbracket \mathbf{m}(\mathcal{I}) \dashv_S \text{State}_S \rrbracket$  then
11        $\text{ctr}_{\text{inc-show}} \leftarrow \text{ctr}_{\text{inc-show}} + 1$ 
12      $\text{State}_S = \text{State}_S \cup \{\mathbf{m}(\mathcal{I})\}$ 
13 return 1

```

Fig. 3.2. Unforgeability experiment

The unforgeability thus covers the model of [BLNS23] through the one-more behavior of step 10 of the game of Figure 3.2, that is we keep track of every inconsistent showings (but possibly allow consistent ones, like showing the same credential twice) and say the adversary wins if it presented more inconsistent showings than corrupt credentials it received. As for the winning condition of step 8, it aims at capturing the unforgeability notion of [FHS19], which is adapted to the fact that attributes may be hidden during issuance. In particular, it ensures that an adversary shall not be able to present a credential with revealed attributes $\mathbf{m}(\mathcal{I})$, if no credential issued to corrupt users can support that. Note that we do not include any restriction related to honest users’ credentials, which means that a credential supporting this presentation may have been issued to an honest user. Our model therefore also captures adversaries that would manage to use a credential without knowing the associated secret key. Finally, one can note that our unforgeability experiment is attribute-oriented, in the sense that it ensures that an adversary cannot claim an attribute it does not have. Whether the adversary uses valid or forged certificates during presentations is irrelevant (and very complex to decide as discussed above) as long as it does not get more power from them.

4 Compact Approximate Range Proofs for LNP

As explained in Section 1, the bulk of the data transmitted when presenting a credential is the zero-knowledge proof of knowledge of the credential and of the concealed attributes, often instantiated with the powerful framework in [LNP22]. In our paper, we however observe that certain components of this framework are suboptimal for the sort of relations we have to prove for anonymous credentials, in particular those related to approximate range proofs. We then revisit the framework of [LNP22] and some of its security proofs by proposing three alternative techniques that can improve its overall performance. We present them in the most general way possible so that they can likely be applied to other instantiations and primitives based on [LNP22].

Let us first describe more formally the parts of [LNP22] that we improve over. We recall that the main attractive feature of the zero-knowledge proof of [LNP22] over previous ones is its ability to naturally handle quadratic lattice relations, with exact Euclidean norm proofs. In our anonymous credentials (and commonly in broader ranges of primitives in lattice cryptography), we have to prove relations of the form $f(\mathbf{w}) = \mathbf{0} \bmod q\mathcal{R}_\pi$, and $\|\mathbf{w}\|_2 \leq B$, where f is a linear or quadratic function. Notice that, at the expense of extending the witness with a single element containing the four-square decomposition of $B^2 - \|\mathbf{w}\|_2^2$, we can reduce the relation to the equality on the norm part. We thus look at the following modified relation

$$f(\mathbf{w}) = \mathbf{0} \bmod q\mathcal{R}_\pi, \quad \text{and} \quad \|\mathbf{w}\|_2^2 = B^2, \quad (1)$$

where $\mathbf{w}$ is of dimension m , and $\mathcal{R}_\pi$ is the power-of-two cyclotomic ring of degree n_π of the proof system³. Note that the norm relation is just an instance of (quadratic) integer relations that will be central in other components of the LNP proof. The latter handles such integer relations by expressing them as evaluations of relations over polynomials. Typically, $\|\mathbf{w}\|_2^2 = B^2$ is equivalent to $\tau_0(\mathbf{w}^*\mathbf{w} - B^2) = 0$, where τ_0 gives the constant coefficient.

However, the proof system natively performs operations and proves relations modulo q_π , where q_π (which may be unrelated to q) must satisfy several arithmetic properties for security and efficiency (prime or product of few primes, adequate splitting of $q_\pi\mathcal{R}_\pi$, $q_\pi - 1$ with a sufficiently large even divisor for compression, etc.). We would then need to prove $f(\mathbf{w}) = q\mathbf{j}$ in $\mathcal{R}_\pi$ for the first part of Equation (1), but through equations modulo q_π . Integer relations like norms should also be handled with care so as to avoid proving meaningless statements due to wraparounds modulo q_π . There is thus a discrepancy between the relations we need to prove and the ones we effectively prove. To address it, the framework of [LNP22] leverages so-called *approximate range proofs* (ARP).

³ Using subring embeddings recalled in Appendix A.1.1, one can also natively tackle relations over larger power-of-two rings $\mathcal{R} \supseteq \mathcal{R}_\pi$. For clarity of presentation, we assume the relations have already been embedded in $\mathcal{R}_\pi$.

4.1 Current Approximate Range Proofs

We describe here the functioning of one ARP component but note that an LNP proof may use several of them [LNP22], to accommodate different constraints more efficiently. We consider a generic vector $\mathbf{x} \in \mathbb{Z}^M$ that we aim at approximately bounding. In practice, this vector is likely to contain $\mathbf{w}$ or the lifting term $\mathbf{j}$ mentioned above, or both.

Sampling Masks and Commitments. The ARP component starts by sampling a mask $\tau(\mathbf{y}_3) \leftarrow \mathcal{D}_{\mathbb{Z}^k, \sigma_3}$ (which defines a vector $\mathbf{y}_3 \in \mathcal{R}_\pi^{k/n_\pi}$) where k is a parameter we discuss below. It then commits to $\mathbf{y}_3$ in the so-called BDLOP part of the commitment, denoted $\mathbf{t}_{B,y}$, as $\mathbf{t}_{B,y} = \mathbf{B}_y \mathbf{s}_2 + \mathbf{y}_3 \bmod q_\pi \mathcal{R}_\pi$. Note that the dimension of $\mathbf{t}_{B,y}$ grows with that of $\mathbf{y}_3$, and it is further increased in the case where the proof involves several parallel ARP components.

To tackle the well-formedness of the ARP that we discuss later (as well as other integer relations) using polynomial relations, the LNP proof also rewrites these integer equations as the zero-evaluation of the constant coefficient of carefully defined polynomials. This relies on the identity that $\langle \tau(\mathbf{a}), \tau(\mathbf{b}) \rangle + c = 0$ is equivalent to $\tau_0(\mathbf{a}^* \mathbf{b} + c) = 0$. The proof then constructs polynomials $f'_1, \dots, f'_\ell$ based on such $\mathbf{a}^* \mathbf{b} + c$ (for each integer relations), while the verifier checks their constant coefficient⁴ is indeed 0. Unfortunately, as is, the other coefficients of the polynomials f'_i may leak information on the witness (through non-constant coefficients of $\mathbf{a}^* \mathbf{b} + c$).

To avoid such leakage, the LNP proof samples a vector $\mathbf{g}$ of polynomials whose coefficients are uniform modulo q_π , except for the constant one (and possibly that of degree $n_\pi/2$ with the technique of [LNP22, Sec. 4.4]) which is 0. The polynomial $\mathbf{g}[i]$ then perfectly masks f'_i in $f_i = \mathbf{g}[i] + f'_i \bmod q_\pi \mathcal{R}_\pi$, while still enabling the verifier check that $\tau_0(f_i) = 0$. It is however a perfect mask only if $\mathbf{g}$ is itself hidden. Moreover, $\mathbf{g}$ must not allow the prover to cheat, which is why the BDLOP commitment above is extended with some $\mathbf{t}_{B,g} = \mathbf{B}_g \mathbf{s}_2 + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi$, giving ℓ additional uniform polynomials of $\mathcal{R}_{\pi, q_\pi}$ in the final proof.

Challenge and Response. The prover then receives a challenge projection matrix $\mathbf{R} \in \{-1, 0, 1\}^{k \times M}$ sampled from a centered binomial distribution (or computes it from a hash when the proof is non-interactive). It then computes the response $\tilde{\mathbf{z}}_3 = \tau(\mathbf{y}_3) + \mathbf{R}\mathbf{x}$. To ensure the zero-knowledge property still holds, the prover must also perform rejection sampling so that $\tilde{\mathbf{z}}_3$ does not leak $\tau(\mathbf{y}_3)$ nor $\mathbf{x}$, which thus enforces a condition on the Gaussian mask parameter σ_3 .

Indeed, as we need to perform (bimodal) Gaussian rejection sampling, we must ensure σ_3 is somewhat larger than $\|\mathbf{R}\mathbf{x}\|_2$, in turn requiring a bound on the latter. This can be obtained from a Johnson-Lindenstrauss bound⁵, giving that $\|\mathbf{R}\mathbf{x}\|_2 \leq \sqrt{\beta^*} \|\mathbf{x}\|_2 \leq \sqrt{\beta^*} B_x$ except with probability at most

⁴ A soundness boosting technique from [LNP22, Sec. 4.4], and implemented in our paper, also leverages the coefficient of degree $n_\pi/2$ to embed other random combinations for the proofs of integer relations. For simplicity, we only discuss the constant coefficient.

⁵ Under the heuristic that the centered binomial distribution behaves as a Gaussian.

$2^{-\lambda}$. From [GHL22,LNP22], we can choose the constant $\beta^* \approx 337$ for $(k, \lambda) = (256, 128)$, or $\beta^* \approx 231$ for $(k, \lambda) = (128, 128)$ for example. Then, for a given rejection rate M_3 , one needs to take $\sigma_3 = \alpha_3 \sqrt{\beta^*} B_x$, with $\alpha_3 = \sqrt{\pi / \ln(M_3)}$ (corresponding to the bimodal case). Such parameters allow for proving that $\tilde{\mathbf{z}}_3$ follows $\mathcal{D}_{\mathbb{Z}^k, \sigma_3}$ after rejection sampling, while incurring (on average) M_3 repetitions.

Well-formedness of the Response. Giving $\tilde{\mathbf{z}}_3$ is not exactly sufficient to deduce a bound on $\mathbf{x}$ because it could have been ill-formed by the prover. One must then add a proof that $\tilde{\mathbf{z}}_3$ is well-formed from the committed values $\mathbf{y}_3 \in \mathcal{R}_\pi^{k/n_\pi}$ and $\mathbf{w}$. As $\mathbf{R}$ is unstructured, this comes down to proving k integer relations of the form $\langle \mathbf{e}_j, \tau(\mathbf{y}_3) \rangle + \langle \mathbf{R}^T \mathbf{e}_j, \tau^{-1}(\mathbf{x}) \rangle - \tilde{\mathbf{z}}_3[j] = 0$, with $\mathbf{e}_j$ the j -th canonical vector. They can be rewritten as $\tau_0(\mathbf{e}_{j, \mathcal{R}_\pi}^* \mathbf{y}_3 + \mathbf{r}_j^* \tau^{-1}(\mathbf{x}) - \tilde{\mathbf{z}}_3[j]) = 0$, for $\mathbf{e}_{j, \mathcal{R}_\pi} = \tau^{-1}(\mathbf{e}_j)$ and $\mathbf{r}_j = \tau^{-1}(\mathbf{R}^T \mathbf{e}_j)$.

Because the $\mathbf{r}_j$ are public, as long as $\mathbf{x}$ is quadratic in the witness $\mathbf{w}$, the well-formedness can be proven in the subsequent components of the LNP proof system, but only modulo q_π . The k polynomials $\mathbf{e}_{j, \mathcal{R}_\pi}^* \mathbf{y}_3 + \mathbf{r}_j^* \tau^{-1}(\mathbf{x}) - \tilde{\mathbf{z}}_3[j]$ defining the above integer equations are then aggregated with random challenge weights uniform in $\mathbb{Z}_{q_\pi}$ (along with polynomials for other integer relations, e.g., $\mathbf{w}^* \mathbf{w} - B^2$ for the equation $\tau_0(\mathbf{w}^* \mathbf{w} - B^2) = 0$), and this random linear combination is used to define the f'_i from above. The f'_i are then masked with the $\mathbf{g}[i]$ as $f_i = \mathbf{g}[i] + f'_i \bmod q_\pi \mathcal{R}_\pi$. The next component of LNP proves the f_i are well-formed as polynomials, and the assurance that all the $\tau_0(f_i)$ are 0 convinces the verifier that the integer relations above are indeed satisfied.

Choice of Norm and Extraction. At this stage, we need to distinguish two different ways of concluding the ARP, depending on whether we want to prove an approximate bound on $\|\mathbf{x}\|_2$ or $\|\mathbf{x}\|_\infty$. For future reference in this text, we specify ARP_2 or ARP_∞ to designate these two types of ARP. Both come with different constraints as we explain here.

First, if one aims at bounding $\|\mathbf{x}\|_2$, the ARP is performed in Euclidean norm, i.e., an ARP_2 . In that case, one sets a verification bound B_{z_3} on the ℓ^2 norm of the masked opening $\tilde{\mathbf{z}}_3$ using the Gaussian tail bound of Lemma 2.2, that is $B_{z_3} = c_k \sigma_3 \sqrt{k}$. When proving soundness and performing the extraction, we then know that the given $\tilde{\mathbf{z}}_3$ is bounded by B_{z_3} and is well-formed as $\tilde{\mathbf{z}}_3 = \tau(\mathbf{y}_3) + \mathbf{R}\mathbf{x}^* \bmod q_\pi \mathbb{Z}$. We can then invoke the *modular* Johnson-Lindenstrauss result of [LNP22, Lem. 2.9] to obtain that the extracted value $\mathbf{x}^*$ satisfies $\|\mathbf{x}^*\|_2 \leq \frac{2}{\sqrt{2\alpha^*}} B_{z_3} =: B_{arp}$, except with probability at most $2^{-\lambda}$. The latter lemma imposes the additional constraints $k = 256$ and $q_\pi > 41MB_{arp}$. The constant α^* (for $(k, \lambda) = (256, 128)$) is $\alpha^* \approx 13$. Overall, it yields the following values for B_{arp} and the bitsize of $\tilde{\mathbf{z}}_3$.

$$B_{arp} = \underbrace{\frac{2}{\sqrt{26}} c_{256} \sqrt{\frac{\pi}{\ln(M_3)}} \sqrt{337} \sqrt{256} \cdot B_x}_{\alpha_{arp}} \quad (2)$$

$$|\tilde{\mathbf{z}}_3| \approx 256 \left(\frac{1}{2} + \log_2 \sigma_3 \right) = 128 \left(1 + \log_2 \left(\frac{\pi}{\ln(M_3)} \cdot 337 \cdot B_x^2 \right) \right). \quad (3)$$

In such an ARP_2 , one can also directly deduce a bound on $\|\mathbf{x}^*\|_\infty$ as $\|\mathbf{x}^*\|_\infty \leq \|\mathbf{x}^*\|_2 \leq B_{arp}$.

Alternatively, for an ARP_∞ , one sets B_{z_3} using the Gaussian tail bound in ℓ^∞ norm from Lemma 2.2, giving $B_{z_3} = t_k \sigma_3$. In the extraction, instead of using [LNP22, Lem. 2.9], we invoke the modular Johnson-Lindenstrauss in ℓ^∞ norm of [LNP22, Lem. 2.7] to obtain that the extracted value $\mathbf{x}^*$ satisfies $\|\mathbf{x}^*\|_\infty \leq 2B_{z_3} =: B_{arp}$, except with probability at most 2^{-k} . The latter does not place any constraint on k nor q_π as opposed to the ARP_2 case. One can typically opt for $k = \lambda = 128$, yielding the following values for B_{arp} and the bitsize of $\tilde{\mathbf{z}}_3$.

$$B_{arp} = \underbrace{2t_{128} \sqrt{\frac{\pi}{\ln(M_3)}} \sqrt{231} \cdot B_x}_{\alpha_{arp}} \quad (4)$$

$$|\tilde{\mathbf{z}}_3| \approx 128 \left(\frac{1}{2} + \log_2 \sigma_3 \right) = 64 \left(1 + \log_2 \left(\frac{\pi}{\ln(M_3)} \cdot 231 \cdot B_x^2 \right) \right). \quad (5)$$

4.2 Relaxing Lifting Proofs

Irrespective of the chosen norm, the ARP component allows for deriving an approximate bound B_{arp} on the concealed vector $\mathbf{x}$ and thus on each of its components, e.g., the lifting term $\mathbf{j}$. As the LNP framework only proves $f(\mathbf{w}) = q\mathbf{j} \bmod q_\pi \mathcal{R}_\pi$, LNP concludes by selecting a modulus q_π that is sufficiently large to ensure no wrap-around occurs. In particular, this means q_π must be at least larger than qB_{arp} , which is quite significant in practice. Our first optimization is to observe that this last condition can be relaxed so as to allow for an additive slack on the extracted lifting equation, slack that would be insignificant in many situations. Going back to Equation (1), let us consider the slightly specified relation

$$[\mathbf{I}_k | \overline{\mathbf{A}}] \mathbf{w} = \mathbf{u} \bmod q \mathcal{R}_\pi, \quad \text{and} \quad \|\mathbf{w}\|_2 = B. \quad (6)$$

The equation corresponds to taking a linear function f but we note that the same reasoning applies to a quadratic one when $\overline{\mathbf{A}}$ may depend on the witness, which is the case in our anonymous credentials. The key specification however is that the matrix has the identity block $\mathbf{I}_k$. To tackle Equation (6), we follow the methodology described in Section 4 by first rewriting the left part as $[\mathbf{I}_k | \overline{\mathbf{A}}] \mathbf{w} = \mathbf{u} + q\mathbf{j}$ in $\mathcal{R}_\pi$. Now, instead of proving the equation holds in $\mathcal{R}_\pi$ by essentially proving $\|[\mathbf{I} | \overline{\mathbf{A}}] \mathbf{w} - \mathbf{u} - q\mathbf{j}\|_\infty < q_\pi$, we relax it by tolerating the latter norm to exceed q_π while remaining bounded by $\xi \cdot q_\pi$ for some relatively small integer ξ . Concretely, it means we proved that $[\mathbf{I} | \overline{\mathbf{A}}] \mathbf{w} = \mathbf{u} + q\mathbf{j} + q_\pi \mathbf{h}$ with $\|\mathbf{h}\|_\infty \leq \xi$. Taking the equation modulo q then gives $[\mathbf{I} | \overline{\mathbf{A}}] \mathbf{w} = \mathbf{u} + (q_\pi \bmod q) \mathbf{h} \bmod q \mathcal{R}_\pi$, which can be written as

$$[\mathbf{I} | \overline{\mathbf{A}}] \underbrace{\left(\mathbf{w} - \begin{bmatrix} (q_\pi \bmod q) \mathbf{h} \\ \mathbf{0} \end{bmatrix} \right)}_{\mathbf{w}'} = \mathbf{u} \bmod q \mathcal{R}_\pi$$

In the soundness, it means we can extract a witness $\mathbf{w}'$ for relation (6), but with a slightly larger norm bound $\|\mathbf{w}'\|_2 \leq B + (q_\pi \bmod q)\xi\sqrt{n_\pi k}$. When the latter extracted witness is leveraged to construct a solution of M-SIS for a matrix linked to $[\mathbf{I}|\bar{\mathbf{A}}]$, we can keep similar security assurances as long as $(q_\pi \bmod q)\xi\sqrt{n_\pi k}$ remains small compared to B . For that, we can tweak q_π to control the value of ξ and in turn the growth of $\mathbf{h}$. The factor $q_\pi \bmod q$ however imposes a certain relation between q and q_π to avoid blowing up the norm. As long as $q_\pi \bmod q$ is small, we can relax the condition of the ARP while ensuring meaningful security without needing q to divide q_π . Typically, we can easily enforce $q_\pi \bmod q = 1$ when selecting parameters so that the additive overhead on the norm of $\mathbf{w}$ is at most $\xi\sqrt{n_\pi k}$.

Note that this departs from the usual approach to build anonymous credentials where the signature scheme and the zero-knowledge framework are treated independently, the former defining relations to prove and the latter being some kind of blackbox able to handle them. Here, we adapt the relations to prove so that they can be more efficiently processed by the LNP framework. This is formally done in Section 5.3. This also affects the security analysis as we need to manage the small discrepancy between the proven relations and the ones underpinning the signature scheme. Fortunately, this does not significantly impact the overall security assurances, as demonstrated in Section 6.

Working Directly in Euclidean Norm. This methodology essentially removes the constraint that q_π should exceed qB_{arp} , at the expense of an additive slack term $(q_\pi \bmod q)\mathbf{h}$. When reducing to M-SIS, we are generally interested in the ℓ^2 norm of the solution, which asks for an ℓ^2 norm bound on $(q_\pi \bmod q)\mathbf{h}$. Following what we described above goes through the ℓ^∞ norm as an intermediate step and essentially gives

$$\begin{aligned} \|\mathbf{h}\|_2 &\leq \frac{1}{q_\pi} (\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_\infty + q\|\mathbf{j}\|_\infty) \sqrt{n_\pi k} \\ &\leq \frac{1}{q_\pi} (\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_\infty + q\|\mathbf{j}\|_2) \sqrt{n_\pi k} \\ &\leq \frac{1}{q_\pi} (\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_\infty + qB_{arp}) \sqrt{n_\pi k}. \end{aligned}$$

Proceeding as such in our anonymous credentials leads to the dominating term being $qB_{arp}\sqrt{n_\pi k}/q_\pi$. The $\sqrt{n_\pi k}$ factor comes from using the ℓ^∞ norm to derive a bound on $\|\mathbf{h}\|_2$.

Here, as opposed to the prior situation when we required a bound in ℓ^∞ norm on $\mathbf{j}$ to argue that every coefficients of the extracted equation remained bounded by q_π , we can now conclude more tightly directly from an ℓ^2 bound on $\mathbf{j}$. This allows us to derive a direct bound on $\|\mathbf{h}\|_2$ as

$$\|\mathbf{h}\|_2 \leq \frac{1}{q_\pi} (\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_2 + q\|\mathbf{j}\|_2) \leq \frac{1}{q_\pi} (\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_2 + qB_{arp}).$$

In our anonymous credentials, and we believe in many other cases, the quantity $\|[\mathbf{I}_k|\bar{\mathbf{A}}]\mathbf{w} - \mathbf{u}\|_2$ remains small before qB_{arp} . It means the dominating term is

now qB_{arp}/q_π , essentially saving a factor of $\sqrt{n_\pi k}$ on the additive slack. Working in ℓ^2 norm then reduces the slack, but it imposes to work with an ARP_2 for (at least) the lifting term $\mathbf{j}$. As mentioned in Section 4.1, ARP_2 s come with additional constraints (on the projection dimension k and the modulus q_π) compared to an ARP_∞ . Our second technique aims at alleviating these specific constraints.

4.3 Refined Modular Johnson-Lindenstrauss Bound

The discussion above highlights the intricacies of the approximate range proofs and the way one can leverage the respective features of ARP_2 and ARP_∞ . Looking at Equations (2) to (5), we make the following observations:

- (1) ARP_∞ allows for choosing a smaller k ($k = 128$ rather than $k = 256$), which reduces the size of $\tilde{\mathbf{z}}_3$ in Equation (5) compared to Equation (3). The reduced dimension also reduces the size of the BDLOP commitment $\mathbf{t}_{B,y}$ as explained in Section 4.1.
- (2) For a typical rejection rate $M_3 = \sqrt{2}$, the value of B_{arp} is smaller for ARP_2 in Equation (2) than for ARP_∞ in Equation (4), by a factor of roughly 2.5. This allows for choosing a slightly smaller q_π .

The loss factor of 2.5 on q_π is vastly compensated by the smaller $\tilde{\mathbf{z}}_3$ and $\mathbf{t}_{B,y}$, which then argues towards an ARP_∞ . The main drawback of an ARP_2 therefore lies in its inability to use $k = 128$ (which would match ARP_∞) and the ad-hoc condition $q_\pi > 41MB_{arp}$. Both are based on how [LNP22, Lem. 2.9] is proven.

We revisit said lemma so as to derive sharper bounds. The main goal is to allow for other choices of k , while also reducing the ad-hoc constraint on q_π . We manage to do both, typically allowing for $k = 128$, while reducing the constraint on q_π to $q_\pi > C_1(k, \lambda)\sqrt{M}B_{arp}$, where M is the dimension of the vector $\mathbf{x}$ involved in the ARP. The constant C_1 is larger than 41, but the condition now scales with the square root of the dimension of $\mathbf{x}$, which is relevant for typical witness dimensions. We prove the following variant of [LNP22, Lem. 2.9]. The latter applies to the coefficient vector of ring elements and we therefore express everything over the integers. We present the lemma in full generality so that it can be applied in place of [LNP22, Lem. 2.9] and not limited to our use-case of lifting equations. Note that this also changes the values β^*, α^* mentioned in Section 4.1. For a more complete view on these constants, we introduce functions $\alpha(k, \lambda), \beta(k, \lambda)$ in Lemma 4.1, and discuss their value for different choices of k later on.

Lemma 4.1. *Let $M, q_\pi, \lambda, k \in \mathbb{N}^\times$, with $k \geq \lambda$, and $\alpha(k, \lambda), \beta(k, \lambda)$ be two functions defined such that for any $\mathbf{x} \in \mathbb{Z}^M$,*

$$\begin{aligned} \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{x}\|_2 < \sqrt{2\alpha(k, \lambda)}\|\mathbf{x}\|_2] &\leq 2^{-\lambda} \\ \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{x}\|_2 > \sqrt{2\beta(k, \lambda)}\|\mathbf{x}\|_2] &\leq 2^{-\lambda}. \end{aligned}$$

We let $\alpha = \alpha(k, 2\lambda + 1)$ and $\beta = \beta(k, 2\lambda + 1)$, and $B_{arp} \leq \frac{q_\pi}{4\sqrt{\alpha\beta M}}$. Then, for all $\mathbf{y} \in \mathbb{Z}_{q_\pi}^k$ and $\mathbf{x} \in (-q_\pi/2, q_\pi/2]^M \cap \mathbb{Z}^M$ such that $\|\mathbf{x}\|_2 \geq B_{arp}$, we have

$$\mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R}\mathbf{x} \bmod q_\pi \mathbb{Z}\|_2 < \frac{1}{2}\sqrt{2\alpha}B_{arp}] \leq 2^{-\lambda}.$$

Proof. To avoid overloading notations, we use B instead of B_{arp} within this proof. We distinguish the following two cases of whether $\|\mathbf{w}\|_2 \leq \frac{q_\pi}{2\sqrt{2\beta}}$ or $\|\mathbf{w}\|_2 > \frac{q_\pi}{2\sqrt{2\beta}}$.

Case 1. We first assume $\|\mathbf{w}\|_2 \leq \frac{q_\pi}{2\sqrt{2\beta}}$. We have the following inequalities.

$$\begin{aligned} \mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R}\mathbf{w} \bmod q_\pi \mathbb{Z}\| < \frac{1}{2}\sqrt{2\alpha}B] \\ &= \mathbb{P}_{\mathbf{R}_1, \mathbf{R}_2 \sim \psi_1^{k \times M}} [\forall i \in \{1, 2\}, \|\mathbf{y} + \mathbf{R}_i \mathbf{w} \bmod q_\pi \mathbb{Z}\| < \frac{1}{2}\sqrt{2\alpha}B]^{1/2} \\ &\leq \mathbb{P}_{\mathbf{R}_1, \mathbf{R}_2 \sim \psi_1^{k \times M}} [\|(\mathbf{R}_1 - \mathbf{R}_2)\mathbf{w} \bmod q_\pi \mathbb{Z}\| < \sqrt{2\alpha}B]^{1/2} \\ &= \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\| < \sqrt{2\alpha}B]^{1/2} \end{aligned}$$

Then, by definition of β , it holds that $\|\mathbf{R}'\mathbf{w}\|_2 \leq \sqrt{2\beta}\|\mathbf{w}\|_2$ except with probability at most $2^{-2\lambda-1}$. Conditioned on this event, we get that $\|\mathbf{R}'\mathbf{w}\|_\infty \leq \sqrt{2\beta}\|\mathbf{w}\|_2 \leq q_\pi/2$. This means there is no wrap-around modulo q_π and thus $\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z} = \mathbf{R}'\mathbf{w}$. As such, $\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 = \|\mathbf{R}'\mathbf{w}\|_2$. Using the definition of α , we then get

$$\begin{aligned} \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \sqrt{2\alpha}\|\mathbf{w}\|_2 \mid \|\mathbf{R}'\mathbf{w}\|_2 \leq \sqrt{2\beta}\|\mathbf{w}\|_2] \\ &= \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{w}\|_2 < \sqrt{2\alpha}\|\mathbf{w}\|_2] \\ &\leq 2^{-2\lambda-1}. \end{aligned}$$

Then, if we denote by E the event $\{\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \sqrt{2\alpha}\|\mathbf{w}\|_2\}$ and $E' = \{\|\mathbf{R}'\mathbf{w}\|_2 \leq \sqrt{2\beta}\|\mathbf{w}\|_2\}$, we have

$$\begin{aligned} \mathbb{P}_{\mathbf{R}'}[E] &= \mathbb{P}_{\mathbf{R}'}[E']\mathbb{P}_{\mathbf{R}'}[E|E'] + \mathbb{P}_{\mathbf{R}'}[\neg E']\mathbb{P}_{\mathbf{R}'}[E|\neg E'] \leq \mathbb{P}_{\mathbf{R}'}[E|E'] + \mathbb{P}_{\mathbf{R}'}[\neg E'] \\ &\leq 2^{-2\lambda-1} + 2^{-2\lambda-1} \\ &= 2^{-2\lambda}. \end{aligned}$$

Then, because $\|\mathbf{w}\|_2 \geq B$, we have

$$\begin{aligned} \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \sqrt{2\alpha}B] \\ &\leq \mathbb{P}_{\mathbf{R}' \sim \psi_2^{k \times M}} [\|\mathbf{R}'\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \sqrt{2\alpha}\|\mathbf{w}\|_2] \\ &< 2^{-\lambda} \end{aligned}$$

Going back to our original calculation, this yields

$$\mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R}\mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \frac{1}{2}\sqrt{2\alpha}B] \leq 2^{-\lambda}$$

which concludes the first case.

Case 2. Let us now assume that $\|\mathbf{w}\|_2 > \frac{q_\pi}{2\sqrt{2\beta}}$. We define $B_\infty = \frac{q_\pi}{4\sqrt{2\beta M}}$. We then first get that $\|\mathbf{w}\|_\infty > \frac{q_\pi}{2\sqrt{2\beta M}} = 2B_\infty$.

We denote by $\mathbf{r}_j^T$ the j -th row of $\mathbf{R}$. Our goal is to show that the probability that $|\mathbf{y}[j] + \mathbf{r}_j^T \mathbf{w} \bmod q_\pi \mathbb{Z}| < B_\infty$ is at most $1/2$ for all $j \in [k]$. Let $i \in [M]$ be the index such that $|\mathbf{w}[i]| = \|\mathbf{w}\|_\infty$, and let $j \in [k]$. For a vector $\mathbf{x} \in \mathbb{Z}^M$, we denote by $\mathbf{x}^{-i}$ the subvector of $\mathbb{Z}^{M-1}$ obtained from $\mathbf{x}$ by removing its i -th entry. We then have

$$\begin{aligned} & \mathbb{P}_{\mathbf{r}_j \sim \psi_1^M} [|\mathbf{y}[j] + \mathbf{r}_j^T \mathbf{w} \bmod q_\pi \mathbb{Z}| < B_\infty] = \\ & \sum_{\mathbf{a} \in [-1, 1]^{M-1}} \mathbb{P}_{\mathbf{r}_j^{-i} \sim \psi_1^{M-1}} [\mathbf{r}_j^{-i} = \mathbf{a}] \mathbb{P}_{r_i \sim \psi_1} [|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} + r_i \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| < B_\infty] \end{aligned}$$

We now look at the second term. Let $\mathbf{a} \in [-1, 1]^{M-1}$, and consider the following cases: (1) $|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} \bmod q_\pi \mathbb{Z}| < B_\infty$, and (2) otherwise.

Case (1). For clarity, we write $z = \mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} \bmod q_\pi \mathbb{Z}$. We show that we necessarily have $|z \pm \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| > B_\infty$. For that, we look at $|z \pm |\mathbf{w}[i]| \bmod q_\pi \mathbb{Z}|$. By assumption of case (1) and the fact that $q_\pi/2 \geq |\mathbf{w}[i]| > 2B_\infty$, we have that $a = z + |\mathbf{w}[i]|$ is in $(B_\infty, B_\infty + q_\pi/2)$. If $a \in (B_\infty, q_\pi/2]$, then $a = a \bmod q_\pi \mathbb{Z}$ and so $a \bmod q_\pi \mathbb{Z} > B_\infty$. On the other hand, if $a \in (q_\pi/2, B_\infty + q_\pi/2)$, then $a \bmod q_\pi \mathbb{Z} = a - q_\pi \in (-q_\pi/2, B_\infty - q_\pi/2)$. As $2B_\infty < q_\pi/2$, we have $B_\infty - q_\pi/2 < -B_\infty$ and therefore $a \bmod q_\pi \mathbb{Z} < -B_\infty$. In both situations, we get $|z + |\mathbf{w}[i]| \bmod q_\pi \mathbb{Z}| > B_\infty$.

Similarly, we have that $a = z - |\mathbf{w}[i]|$ lies in $(-B_\infty - q_\pi/2, -B_\infty)$. If $a \in (-q_\pi/2, -B_\infty)$, then $a \bmod q_\pi \mathbb{Z} = a < -B_\infty$. And if $a \in (-B_\infty - q_\pi/2, -q_\pi/2]$, then $a \bmod q_\pi \mathbb{Z} = a + q_\pi \in (q_\pi/2 - B_\infty, q_\pi/2] \subseteq (B_\infty, q_\pi/2]$. In any case, $|z - |\mathbf{w}[i]| \bmod q_\pi \mathbb{Z}| > B_\infty$. As $\{\pm \mathbf{w}[i]\} = \{\pm |\mathbf{w}[i]|\}$, this indeed proves that $|z \pm \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| > B_\infty$. As a result, we get

$$\begin{aligned} & \mathbb{P}_{\mathbf{r}_j[i] \sim \psi_2} [|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} + \mathbf{r}_j[i] \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| < B_\infty] \\ &= \sum_{r_i \in [-1, 1]} \mathbb{P}_{\mathbf{r}_j[i] \sim \psi_1} [\mathbf{r}_j[i] = r_i] \cdot \mathbf{1}(|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} + r_i \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| < B_\infty) \\ &= \mathbb{P}_{\mathbf{r}_j[i] \sim \psi_1} [\mathbf{r}_j[i] = 0] \\ &= \frac{1}{2}. \end{aligned}$$

Case (2). Then, by definition of case (2), it holds that $|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} \bmod q_\pi \mathbb{Z}| \geq B_\infty$ (corresponding to $r_i = 0$). Hence

$$\begin{aligned} & \mathbb{P}_{\mathbf{r}_j[i] \sim \psi_1} [|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} + \mathbf{r}_j[i] \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| < B_\infty] \\ &\leq \sum_{r_i \in [-1, 1] \setminus \{0\}} \mathbb{P}_{\mathbf{r}_j[i] \sim \psi_1} [\mathbf{r}_j[i] = r_i] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}. \end{aligned}$$

Overall, we have $\mathbb{P}_{\mathbf{r}_j[i] \sim \psi_1} [|\mathbf{y}[j] + \mathbf{a}^T \mathbf{w}^{-i} + \mathbf{r}_j[i] \mathbf{w}[i] \bmod q_\pi \mathbb{Z}| < B_\infty] \leq 1/2$. Going back to our original computation, this shows

$$\mathbb{P}_{\mathbf{r}_j \sim \psi_1^M} [|\mathbf{y}[j] + \mathbf{r}_j^T \mathbf{w} \bmod q_\pi \mathbb{Z}| < B_\infty] \leq \sum_{\mathbf{a} \in [-1,1]^{M-1}} \mathbb{P}_{\mathbf{r}_j^{-i} \sim \psi_1^{M-1}} [\mathbf{r}_j^{-i} = \mathbf{a}] \cdot \frac{1}{2} = \frac{1}{2}$$

Thence, we get $\mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R} \mathbf{w} \bmod q_\pi \mathbb{Z}\|_\infty < B_\infty] \leq 2^{-k} \leq 2^{-\lambda}$. Using $\|\cdot\|_\infty \leq \|\cdot\|_2$, we then get

$$\begin{aligned} \mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R} \mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < B_\infty] \\ \leq \mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R} \mathbf{w} \bmod q_\pi \mathbb{Z}\|_\infty < B_\infty] \\ \leq 2^{-\lambda} \end{aligned}$$

Finally, as $\frac{1}{2} \sqrt{2\alpha} B \leq B_\infty$, we get

$$\mathbb{P}_{\mathbf{R} \sim \psi_1^{k \times M}} [\|\mathbf{y} + \mathbf{R} \mathbf{w} \bmod q_\pi \mathbb{Z}\|_2 < \frac{1}{2} \sqrt{2\alpha} B] \leq 2^{-\lambda},$$

which concludes case 2. Hence, for either case 1 and case 2, we have proven the desired result, which yields the claim. $\square$

Lemma 4.1 then states that if the Johnson-Lindenstrauss bounds hold with a given probability, we can deduce a modular version when q_π satisfies the condition $q_\pi > \Omega(B_{arp} \sqrt{M})$. We are then left with evaluating the constants α, β . Using a similar heuristics as in [GHL22, LNP22] that $\|\mathbf{R} \cdot \mathbf{1}_M\|_2^2$ follows $\frac{\eta}{2} M \chi^2[k]$ for $\mathbf{R}$ following ψ_η , we get that

$$\begin{aligned} \mathbb{P}_{\mathbf{R} \sim \psi_\eta^{k \times M}} [\|\mathbf{R} \mathbf{x}\|_2 < \sqrt{\eta \alpha} \|\mathbf{x}\|_2] &\lesssim \mathbb{P}_{x \sim \chi^2[k]} [x < 2\alpha] = cdf_{\chi^2[k]}(2\alpha) \\ \mathbb{P}_{\mathbf{R} \sim \psi_\eta^{k \times M}} [\|\mathbf{R} \mathbf{x}\|_2 > \sqrt{\eta \beta} \|\mathbf{x}\|_2] &\lesssim \mathbb{P}_{x \sim \chi^2[k]} [x > 2\beta] = sf_{\chi^2[k]}(2\beta) \end{aligned}$$

where $cdf_{\chi^2[k]}, sf_{\chi^2[k]}$ are the cumulative distribution function and survival function respectively of the χ^2 distribution with k degrees of freedom. We can then evaluate the functions α, β at each given pair (k, λ) as $\alpha(k, \lambda) = cdf_{\chi^2[k]}^{-1}(2^{-\lambda})/2$ and $\beta(k, \lambda) = sf_{\chi^2[k]}^{-1}(2^{-\lambda})/2$. In our case, we need to use Lemma 4.1 with $\lambda = 128$, but we are free to choose whichever $k \geq \lambda$ we want as opposed to [LNP22]. This provides a tradeoff between the dimension of $\mathbf{R}$ (and thus the masked opening $\tilde{\mathbf{z}}_3$) and the slacks α, β .

The parameter $C_1(k, \lambda) = 4\sqrt{\alpha(k, 2\lambda + 1)\beta(k, 2\lambda + 1)}$ translates to the condition the modulus q_π must satisfy with respect to the bound B_{arp} and the dimension M as $q_\pi > C_1(k, \lambda) \sqrt{M} B_{arp}$. This contrasts with the condition $q_\pi > 41M B_{arp}$ from ARP_2 which scales linearly with the witness dimension. As shown in Table 4.1, we always have $C_1 \geq 41$. Typically, for $k = 256$ as in [LNP22], C_1 is about 7.7 times larger than in [LNP22]. But the constant being larger is compensated by the extra $\sqrt{M}$ factor of [LNP22] for typical values of M .

The choice of k also affects the bound on the extracted value of $\mathbf{x}$ from the approximate range proof. As we have seen in Section 4.1, one must first have a bound $\|\mathbf{R}\mathbf{x}\|_2$ that is satisfied with probability at least $1 - 2^{-\lambda}$ to perform rejection sampling securely. Using the function β , we have $\|\mathbf{R}\mathbf{x}\|_2 \leq \sqrt{\beta(k, \lambda)} \|\mathbf{x}\|_2 \leq \sqrt{\beta(k, \lambda)} B_x$, where B_x is an a priori bound on the $\mathbf{x}$. This allows for defining the Gaussian parameter of the mask $\mathbf{y}_3$ as $\sigma_3 = \alpha_3 \sqrt{\beta(k, \lambda)} B_x$, where $\alpha_3 = \sqrt{\pi / \ln(M_3)}$. Then, the verification bound on $\tilde{\mathbf{z}}_3$ is defined as $B_{z_3} = c_k \sigma_3 \sqrt{k}$. From this bound, Lemma 4.1 then gives, during the extraction, that $\|\mathbf{w}\|_2 \leq B_{z_3} \cdot 2 / \sqrt{2\alpha(k, 2\lambda + 1)}$. Putting everything together, compared to the initial bound B_x , the bound obtained from the approximate range proof is

$$B_{arp} = \alpha_3 \cdot \underbrace{\frac{2c_k \sqrt{k \cdot \beta(k, \lambda)}}{\sqrt{2\alpha(k, 2\lambda + 1)}}}_{C_2(k, \lambda)} \cdot B_x$$

The choice of k impacts the value of $C_2(k, \lambda)$, then determining the quality of the approximate range proof. We give different choices of k and the associated slacks below.

k	$\alpha(k, 2\lambda + 1)$	$\beta(k, 2\lambda + 1)$	$\beta(k, \lambda)$	$C_1(k, \lambda)$	$C_2(k, \lambda)$
128	1.56263	345.57891	230.86567	92.95262	149.09047
192	6.07834	408.85694	285.44437	199.40600	93.31684
256	13.32599	467.66593	336.66396	315.77493	74.25522

Table 4.1. Modular Jonhson-Lindenstrauss slacks for different values of k and $\lambda = 128$.

When $k = 256$, this matches the result of [LNP22, Lem. 2.9], albeit with a different ad-hoc condition scaling with the square root rather than linearly. However, when taking $k = 128$, we observe that $\alpha_3 C_2$ incurs an overhead factor of 2 compared to the $k = 256$ case of Equation (2). Compared with an ARP_∞ above where this overhead factor was of 2.5 in Equation (4), our method for ARP_2 with Lemma 4.1 then outperforms ARP_∞ on all aspects. This overhead factor of 2 is then compensated by the reduced dimension which improves the sizes of $\tilde{\mathbf{z}}_3$ and $\mathbf{t}_{B,y}$.

This improvement of ARP_2 s then makes them perfectly suited for the relaxed lifting proofs in ℓ^2 norm mentioned in Section 4.2. Additionally, because lifting ARPs are usually more demanding than ARPs used to prove $\mathbf{w}$ is short enough so that $\|\mathbf{w}\|_2^2 = B^2$, we typically want to use two independent ARPs. One for the witness, yielding a bound $B_{arp,1}$ and a condition $q_\pi > O(B_{arp,1}^2)$, and one for the lifting that proves $\|\mathbf{j}\|_2 \leq B_{arp,2}$ and thus $\|\mathbf{h}\|_2 \leq O(q B_{arp,2} / q_\pi)$. This avoids requiring q_π to scale with B_j^2 (B_j being an a priori bound on $\|\mathbf{j}\|_2$), which would mean the lifting would still impact q_π despite our relaxed lifting technique. Thanks to Lemma 4.1 and the reduction of the projection dimension

k , we figuratively have decoupled these two ARPs for the same price as the original ARP from [LNP22], but with a much smaller modulus q_π .

4.4 Implicit Garbage Polynomials

As described in Section 4.1, integer relations (including the well-formedness of the ARP masked openings $\tilde{\mathbf{z}}_3$) are handled with random linear combinations of polynomials whose coefficients of degree 0 (and $n_\pi/2$ with the soundness boosting technique of [LNP22, Sec. 4.4]) evaluate to 0. These random combinations f'_i are masked by $\mathbf{g}[i]$ which is uniform in the set $\{g \in \mathcal{R}_{\pi, q_\pi} : \tau_0(g) = \tau_{n_\pi/2}(g) = 0\}$. Said mask vector $\mathbf{g}$ must however remain hidden to the verifier, and committed to before receiving the challenge c , which is why it is committed to in the BDLOP part $\mathbf{t}_{B,g} = \mathbf{B}_g \mathbf{s}_2 + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi$. This yields uniform elements in $\mathcal{R}_{\pi, q_\pi}$ whose number grows with the dimension ℓ of $\mathbf{g}$.

As explained in the introduction, our main observation here is that we are actually concealing a mask, $\mathbf{g}$, with an other mask, $\mathbf{B}_g \mathbf{s}_2$, which does not look optimal. Our key idea is then to rely on the entropy of $\mathbf{B}_g \mathbf{s}_2$ to build $\mathbf{g}$, which requires some caution. Removing $\mathbf{t}_{B,g}$ altogether and directly using $\mathbf{B}_g \mathbf{s}_2$ to mask the f'_i is not that simple as we need to project $\mathbf{B}_g \mathbf{s}_2$ on the set of polynomials whose coefficients of degree 0 and $n_\pi/2$ are 0. While this projection may seem harmless, it creates some issues in the soundness proof as we still need to ensure that the resulting polynomials are correctly derived from the witness $\mathbf{s}_2$. Since $\mathbf{s}_2$ is not directly used in the proof but only through its masked version $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$, it creates some challenges as (1) the projection of $c\mathbf{s}_2$ is not the product of the respective projections and (2) the projection now affects $\mathbf{y}_2$ and in turn several other components of the LNP proofs, which drastically changes the soundness analysis.

We thus proceed differently and generate $\mathbf{t}_{B,g} = \mathbf{B}_g \mathbf{s}_2 + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi$ as usual except that we define each component $\mathbf{g}[i]$ of $\mathbf{g}$ as $-(\mathbf{B}_g \mathbf{s})[i]$ where we zeroized its coefficients of degree 0 and $n_\pi/2$. We thus define the function $\text{proj} : r \in \mathcal{R}_\pi \mapsto r - \tau_0(r) - \tau_{n_\pi/2}(r)x^{n_\pi/2}$, and extend it to vectors over $\mathcal{R}_\pi$ entrywise. Then, we define

$$\mathbf{g} = -\text{proj}(\mathbf{B}_g \mathbf{s}_2 \bmod q_\pi \mathcal{R}_\pi) = \left[- \sum_{j \neq 0, n_\pi/2} \tau_j((\mathbf{B}_g \mathbf{s}_2 \bmod q_\pi \mathcal{R}_\pi)[i]) x^j \right]_{i \in [\ell]}$$

The main advantage of this technique, beyond simplifying the protocols and avoiding the sampling of a fresh mask $\mathbf{g}$, is the compression of the commitment $\mathbf{t}_{B,g}$. Indeed, as it now lies in the set of vectors of polynomials with zero coefficients except those of degree 0 and $n_\pi/2$, each entry of $\mathbf{t}_{B,g}$ only requires sending 2 coefficients within the final proof. We thus compress the size of $\mathbf{t}_{B,g}$ from $\ell \cdot n_\pi \lceil \log_2 q_\pi \rceil$ to $\ell \cdot 2 \lceil \log_2 q_\pi \rceil$, a $n_\pi/2 = 32\times$ improvement over the original one. Moreover, this approach has the merit of having no impact on correctness and on soundness as $\mathbf{g}$ retains its key features. It thus only remains to check that the zero-knowledge property is unaffected. This is formally done in Ap-

pendix A for the relations of our anonymous credentials, but we provide the core arguments below.

Concretely, in the security proof of [LNP22] for the zero-knowledge property (also formally described in [AGJ⁺24, Lem. 7.3]), adapted to the notations of our protocol, the challenger generates all the commitments involving $\mathbf{s}_2$ as random elements and argue indistinguishability from the previous game based on the fact that the underlying masks are of the form $\mathbf{B}\mathbf{s}_2$, which are themselves indistinguishable from random elements under (knapsack) M-LWE. Note that $\mathbf{B}$ is in fact the concatenation of $\mathbf{B}_y, \mathbf{B}_g$ and possibly other parts of the ABDLOP commitment.

We proceed exactly in the same way in our case and, given an instance $\mathbf{u}$ of the knapsack version of M-LWE, we let $\mathbf{u}_g$ be the component corresponding to $\mathbf{B}_g$ and generate, for $i \in [\ell]$:

$$\mathbf{g} = -\text{proj}(\mathbf{u}_g), \text{ and } \mathbf{t}_{B,g} = \mathbf{u}_g + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi = \left[\tau_0(\mathbf{u}_g[i]) + \tau_{n_\pi/2}(\mathbf{u}_g[i])x^{n_\pi/2} \right]_{i \in [\ell]}$$

In the case where these elements are built from a short vector $\mathbf{s}_2$, this is exactly our protocol specification. If $\mathbf{u}$, and thus $\mathbf{u}_g$, is uniform, we are ensured that $\mathbf{g}$ is uniform in $\{g \in \mathcal{R}_{\pi, q_\pi} : \tau_0(g) = \tau_{n_\pi/2}(g) = 0\}^\ell$ and independent of $\mathbf{t}_{B,g}$, which is itself uniform in $\{r + r'x^{n_\pi/2}; (r, r') \in \mathbb{Z}_{q_\pi}^2\}^\ell$. The rest of the proof of the zero-knowledge property then falls back on the same arguments as in [LNP22, AGJ⁺24]. This readily gives a zero-knowledge property based on the hardness of (knapsack) M-LWE, as in the original LNP proof system.

5 An Improved Anonymous Credentials

5.1 High-Level Description

Our contributions presented above target the zero-knowledge proof component of anonymous credentials but do not change the overall structure of such systems. As a consequence, our construction follows the classical framework where the users get signatures on their attributes during **Issue** and then prove knowledge of them (and of the concealed attributes) during **Show**.

For the signature scheme, we choose to implement the one in [JS26] which is, to our knowledge, the most efficient standard model construction proven under well-known lattice assumptions. It works with a modulus $q = q_L q_H$ where $q_L = b_L^{k_L}$ and $q_H = b_H^{k_H}$, for some integers b_L and b_H . Note that this decomposition is not ideal for standard LNP proofs, but this is no longer a problem thanks to our results. In practice, a signature on a vector $\mathbf{m}$ is a short vector $\mathbf{w}$ such that

$$[\mathbf{I}|\mathbf{A}|q_L t \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{w} = \mathbf{u} + \mathbf{D}\mathbf{m} \bmod q\mathcal{R}$$

where $\mathbf{A}$, $\mathbf{D}$, $\mathbf{a}$ and $\mathbf{u}$ are public elements without any expected structure (e.g. they can be generated as hash outputs), $\mathbf{G}_H$ is a gadget matrix, t is some tag, and $\mathbf{B} = [\mathbf{I}|\mathbf{A}]\mathbf{R}$ for some short matrix $\mathbf{R}$ acting as the signer's trapdoor.

In our system, $\mathbf{m}$ is replaced by the vector of attributes to sign, but it leaves open the question of the user secret key. As in [JS26], it could be processed as

any other attribute, but it seems suboptimal as it serves a different purpose. We indeed recall that its only goal is to ensure that the signature is useless to any adversary without access to this key, even one with knowledge of all the signed attributes. If $[\mathbf{s}_1^T | \mathbf{s}_2^T]^T$ is the user secret key, we can then add $\mathbf{s}_1 + \mathbf{A}\mathbf{s}_2$ to the right-hand side of the equation which, eventually, allows for merging $[\mathbf{s}_1^T | \mathbf{s}_2^T]^T$ with $\mathbf{w}$. This reduces the size of the witness in the zero-knowledge proof, without jeopardizing security, as we demonstrate in our security analysis.

If each user were limited to one credential per key, then we could have used this secret key to hide the component $\mathbf{D}\mathbf{m}$, and hence the set of signed attributes. However, our model supports several issuances and thus requires an additional source of entropy. This is done by adding $\mathbf{r}_1 + \mathbf{A}\mathbf{r}_2$ for some short vector $\mathbf{r}_1, \mathbf{r}_2$ that is also merged with $\mathbf{w}$, thereby minimizing the impact on the credential.

Once the credential has been issued, the user is able to present them by hiding all the necessary elements in the zero-knowledge proof, which can be done much more efficiently thanks to our contributions.

5.2 Specifications

We now give the detailed specifications of our anonymous credentials scheme. Below, the sampler uses the ring $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ while the zero-knowledge proofs π_{iss} and π are computed in the subring $\mathcal{R}_\pi = \mathbb{Z}[x]/\langle x^{n_\pi} + 1 \rangle$, with $n_\pi | n$.

Algorithm 5.1: Setup(1^λ)

1. Choose $n \geq n_\pi \geq \kappa$ to be powers of two, and define $k_\pi \leftarrow n/n_\pi$
2. Choose $\omega \in \mathbb{N}^\times$ s.t. $\binom{n_\pi}{\omega} \geq Q_{\text{iss}}$
3. $\mathcal{T}_\omega \leftarrow \{t \in T_1 : t = r(x^{k_\pi}) \text{ for some } r \in \mathcal{R}_\pi \text{ with } \tau(r) \in \{0, 1\}^{n_\pi} \text{ and } \|r\|_1 = \omega\}$
- Parameters of TSampler.**
4. Choose $d \in \mathbb{N}^\times$
5. Choose $m \in \mathbb{N}^\times$ $\triangleright$ number of attributes
6. Choose $b_H \in \mathbb{N}^\times$ to be prime s.t. $b_H = 2\kappa + 1 \pmod{4\kappa}$ and $b_H > \sqrt{\kappa}^\kappa$
7. Choose $k_H, b_L, k_L \in \mathbb{N}^\times$
8. $q_L \leftarrow b_L^{k_L}$
9. $q_H \leftarrow b_H^{k_H}$
10. $q \leftarrow q_L q_H$
11. $B_R \leftarrow \frac{3}{4}(\sqrt{nd} + \sqrt{ndk_H} + 6)$
12. For $x \in \{L, H\}$, define $\mathbf{G}_x = [1 \dots |b_x^{k_x-1}|] \otimes \mathbf{I}_d \in \mathcal{R}^{d \times dk_x}$
13. Choose $\varepsilon \in (0, 1)$ $\triangleright$ typically $\varepsilon \approx 2^{-40}$
14. $s_L \leftarrow b_L \cdot \sqrt{\ln(2ndk_L/\varepsilon)/\pi}$ $\triangleright s_L \approx b_L \eta_\varepsilon(\mathbb{Z}^{ndk_L}) \geq b_L \eta_{\varepsilon/dk_L}(\mathbb{Z}^n)$
15. $s_H \leftarrow b_H \cdot \sqrt{\ln(2ndk_H/\varepsilon)/\pi}$ $\triangleright s_H \approx b_H \eta_\varepsilon(\mathbb{Z}^{ndk_H}) \geq b_H \eta_{\varepsilon/dk_H}(\mathbb{Z}^n)$
16. $\eta \leftarrow \sqrt{\ln(2nd(k_L + k_H)/\varepsilon)/\pi}$ $\triangleright \eta \approx \eta_\varepsilon(\mathbb{Z}^{nd(k_L + k_H)})$
17. $s_1 \leftarrow \sqrt{\frac{s_L^4}{s_L^2 - \eta^2} \cdot \frac{b_L^{2k_L-1}}{b_L^2 - 1} + 3 \frac{s_H^4}{s_H^2 - \eta^2} B_R^2}$
18. $s_2 \leftarrow \sqrt{3} \cdot \frac{s_H^2}{\sqrt{s_H^2 - \eta^2}} B_R$
19. $s_3 \leftarrow \sqrt{3} \cdot \frac{s_H^2}{\sqrt{s_H^2 - \eta^2}}$
20. Choose $\eta_r \in \mathbb{N}^\times$
21. $B_r \leftarrow K \sqrt{nd\eta_r/2}$ $\triangleright K \approx 1.15$ (binomial tailcut)
22. $\alpha_1 \leftarrow s_1 / (cn\sqrt{md/2} + B_r + \sqrt{nd} + B_{h,\text{iss},1} + B_{h,\text{iss},2})$ $\triangleright c \approx 1.061$ [DFJS26]
23. $\alpha_2 \leftarrow s_2 / (cn\sqrt{md/2} + B_r + \sqrt{nd})$

24. $M_i \leftarrow \exp(\pi/\alpha_i^2)$ for $i \in \{1, 2\}$
 25. $\text{params} \leftarrow (n, d, m, b_H, b_L, k_H, k_L, q_H, q_L, q, w, B_R, s_H, s_L, \eta, s_1, s_2, s_3, \eta_r, B_r, M_1, M_2)$
 26. $\text{st} \leftarrow 1$ $\triangleright$ It could also be sampled from $U(\{0, 1\}^{256})$
Expanded Public Elements.
 27. $\text{seed} \leftarrow U(\{0, 1\}^{256})$
 28. $\mathbf{A} \leftarrow \mathcal{H}(\text{seed}, \text{dsep}_1)$ $\triangleright \mathbf{A} \sim U(\mathcal{R}_q^{d \times d})$
 29. $\mathbf{u} \leftarrow \mathcal{H}(\text{seed}, \text{dsep}_2)$ $\triangleright \mathbf{u} \sim U(\mathcal{R}_q^d)$
 30. $\mathbf{a} \leftarrow \mathcal{H}(\text{seed}, \text{dsep}_3)$ $\triangleright \mathbf{a} \sim U(\mathcal{R}_q^d)$
 31. $\mathbf{D} \leftarrow \mathcal{H}(\text{seed}, \text{dsep}_4)$ $\triangleright \mathbf{D} \sim U(\mathcal{R}_q^{d \times m})$
Output: $\text{pp} = (\text{params}, \text{seed})$

Algorithm 5.2: OKeyGen(pp)

1. $\mathbf{R}_1, \mathbf{R}_2 \leftarrow \mathcal{B}_1^{d \times dk_H}$ conditioned on $\|\mathbf{R}_i\|_2 \leq B_R$
2. $\mathbf{B} \leftarrow \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2 \bmod q\mathcal{R}$

Output: $\text{opk} = \mathbf{B}, \text{osk} = (\mathbf{R}_1, \mathbf{R}_2)$

Algorithm 5.3: UKeyGen(pp)

1. $\mathbf{s}_1, \mathbf{s}_2 \leftarrow U(T_1^d)$
2. $\mathbf{b} \leftarrow \mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 \bmod q\mathcal{R}$

Output: $\text{upk} = \mathbf{b}, \text{usk} = (\mathbf{s}_1, \mathbf{s}_2)$

Algorithm 5.4: Issue($O(\text{osk}, \text{opk}, \text{upk}, \text{pp}, \mathbf{m}(\mathcal{I}), \text{st}), U(\text{usk}, \text{upk}, \text{opk}, \text{pp}, \mathbf{m}, \mathcal{I})$)

User U .

1. $\mathbf{r}_1, \mathbf{r}_2 \leftarrow \mathcal{B}_{\eta_r}^d$ conditioned on $\|\mathbf{r}_i\|_2 \leq B_r$
2. $\mathbf{c} \leftarrow \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \text{upk} + \mathbf{D}\mathbf{m} \bmod q\mathcal{R}$
3. $\pi_{\text{iss}} \leftarrow \text{Prove}_{\text{iss}}(\text{statement}_{\text{iss}}, (\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}(\bar{\mathcal{I}})))$
4. Send $(\mathbf{c}, \pi_{\text{iss}})$ to O .
5. Organization O .
6. **if** $\llbracket \text{Verify}_{\text{iss}}(\text{statement}_{\text{iss}}, \pi_{\text{iss}}) = 0 \rrbracket$ **abort**
7. $t \leftarrow F(\text{st}) \in \mathcal{T}_\omega$
8. $\text{st} \leftarrow \text{st} + 1$
9. **do**
10. $\mathbf{v} \leftarrow \text{TSampler}(\mathbf{R}_1, \mathbf{R}_2, \mathbf{A}, \mathbf{u} + \mathbf{c} \bmod q\mathcal{R}, t, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_L, \mathbf{s}_H)$
11. **while** $\llbracket \|\mathbf{v}_1\|_2 > B_1 \rrbracket \vee \llbracket \|\mathbf{v}_2\|_2 > B_2 \rrbracket \vee \llbracket \|\mathbf{v}_3\|_2 > B_3 \rrbracket$
12. Send $(t, \mathbf{v}_2, \mathbf{v}_3)$ to U .
13. User U .
14. $\mathbf{v}_1 \leftarrow \mathbf{u} + \mathbf{c} - \mathbf{A}\mathbf{v}_2 - [q_L t \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3 \bmod^\pm q\mathcal{R}$
15. **if** $\llbracket \|\mathbf{v}_1\|_2 > B_1 \rrbracket \vee \llbracket \|\mathbf{v}_2\|_2 > B_2 \rrbracket \vee \llbracket \|\mathbf{v}_3\|_2 > B_3 \rrbracket \vee \llbracket t \notin \mathcal{T}_\omega \rrbracket$ **abort**
16. $\mathbf{v}'_2 \leftarrow \mathbf{v}_2 - \mathbf{r}_2$

Output: O gets $\text{cred} = (t, \mathbf{v}_2, \mathbf{v}_3)$, and U gets $\text{cred}_{\text{unb}} = (t, \mathbf{v}'_2, \mathbf{v}_3)$

Algorithm 5.5: Show($U(\text{usk}, \text{upk}, \text{opk}, \text{pp}, \mathbf{m}, \mathcal{I}, \text{cred}_{\text{unb}}), V(\text{opk}, \text{pp}, \mathbf{m}(\mathcal{I}))$)

User U .

1. $\mathbf{w}_2 \leftarrow \mathbf{v}'_2 - \mathbf{s}_2$
2. $\mathbf{w}_1 \leftarrow \mathbf{u} + \mathbf{D}\mathbf{m} - \mathbf{A}\mathbf{w}_2 - [q_L t \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3 \bmod^\pm q\mathcal{R}$
3. $\pi \leftarrow \text{Prove}(\text{statement}, (\mathbf{w}_1, \mathbf{w}_2, \mathbf{v}_3, t, \mathbf{m}(\bar{\mathcal{I}})))$
4. Verifier V .
5. $\rho \leftarrow \llbracket \text{Verify}(\pi, \text{statement}) = 1 \rrbracket$

Output: ρ (1 if valid, 0 otherwise)

The verification bounds on the preimage sent by the issuer, i.e., B_1, B_2, B_3 , are set using the Gaussian tail bound of Lemma 2.2. More precisely, $B_i = c_{\dim(\tau(\mathbf{v}_i))} s_i \sqrt{\dim(\tau(\mathbf{v}_i))}$, with c_N the Gaussian tailcut giving a tail probability of at most p . To avoid too many repetitions, we take $p = 2^{-20}$.

The function F is an injective map from $[1, |\mathcal{T}_\omega|]$ to $\mathcal{T}_\omega$ so as to ensure that O will never generate two credentials with the same tag.

Remark 5.1. For sake of simplicity, we describe the general issuance case where $\mathbf{c}$ is generated as a hiding commitment. In the specific scenario where no attribute is concealed, that is $\mathcal{I} = [m]$, there is no need to hide $\mathbf{m}$ and one can then select $\mathbf{r}_1 = \mathbf{r}_2 = \mathbf{0}$. In that case, cred and cred_{unb} match.

We define the unblinded credential cred_{unb} to be the certificate produced by the issuer *after* merging the commitment randomness but *before* merging the user secret key. Functionally, this avoids having to store the randomness $\mathbf{r}_2$ and merge it at every presentation of the credential. This also allows us to be coherent with the security model where the adversary may learn the cred_{unb} but should not have the ability to present it without the honest user's secret key. If we were to merge $\mathbf{s}_2$ in cred_{unb} , then knowing cred would suffice to present it. This subtlety in vocabulary on what we really call “the (unblinded) credential” highlights the peculiarities of device-binding and non-transferability of credentials. Here, the unblinded credential may be transfered, but cannot be presented without the user secret key, which, in practice, would be stored on a secure element of the device, in turn tying the presentation ability to the device.

5.3 Zero-Knowledge Relations

Let us now describe formally the relations that are proven with $(\text{Prove}_{\text{iss}}, \text{Verify}_{\text{iss}})$ and $(\text{Prove}, \text{Verify})$. A complete description of the protocols themselves can be found in Appendix A for completeness. In particular, we describe how to seamlessly incorporate our optimizations within the LNP framework of [LNP22].

First, the issuance of credentials requires the user to prove the opening of its commitment as well as the knowledge of the user secret key underlying upk . In doing so, the user has the freedom to disclose or hide all or part of its attributes. Decomposing the matrix $\mathbf{D}$ as $\mathbf{D} = [\mathbf{d}_1 | \dots | \mathbf{d}_m]$ allows for accomodating this selective disclosure in the proven relation. With our technique of Section 4.2, the equation is actually proven with a small additive slack $(q_\pi \bmod q)\mathbf{h}_{\text{iss}}$ that appears in the soundness proof. Although it does not prove the equation exactly, we show the security analysis still goes through with only a very mild impact on the M-SIS bound. Overall, the user proves knowledge of $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}(\bar{\mathcal{I}})) \in \mathcal{R}^{4d+m-|\mathcal{I}|}$ such that there exists some $\mathbf{h}_{\text{iss},1}, \mathbf{h}_{\text{iss},2} \in \mathcal{R}^d$ with $\|\mathbf{h}_{\text{iss},i}\|_2 \leq B_{h,\text{iss},i}$ for $i \in \{1, 2\}$ that verify

$$\begin{aligned} \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \sum_{i \in [m] \setminus \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\bar{\mathcal{I}})[i] - (q_\pi \bmod q)\mathbf{h}_{\text{iss},1} \\ = \mathbf{c} - \text{upk} - \sum_{i \in \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\mathcal{I})[i] \bmod q\mathcal{R} \\ \mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 - (q_\pi \bmod q)\mathbf{h}_{\text{iss},2} = \text{upk} \bmod q\mathcal{R} \\ \|\mathbf{r}_i\|_2 \leq B_r \text{ for } i \in \{1, 2\}, \quad \mathbf{s}_i \in T_1^d \text{ for } i \in \{1, 2\}, \text{ and } \forall i \in [m] \setminus \mathcal{I}, \mathbf{m}(\bar{\mathcal{I}})[i] \in T_1 \end{aligned}$$

As noted in the remark above, in the case where no attribute is concealed, we have $\bar{\mathcal{I}} = \emptyset$ and $\mathbf{r}_1 = \mathbf{r}_2 = \mathbf{0}$, which means that the proof can be significantly simplified as only $\mathbf{s}_1$ and $\mathbf{s}_2$ remain in the list of witnesses.

Then, whenever the user wants to anonymously present a credential to a relying party through the **Show** protocol, it proves knowledge of the validity of its credential as a signature on its hidden attributes. Here again, we apply our relaxed lifting technique of Section 4.2 which incurs a small additive slack $(q_\pi \bmod q)\mathbf{h}$. The term $\mathbf{h}$ from the extracted **Show** proof is taken into account in the security proof, yielding only a negligible overhead on the M-SIS bound. Overall, the user proves knowledge of $(\mathbf{w}_1, \mathbf{w}_2, \mathbf{v}_3, t, \mathbf{m}(\bar{\mathcal{I}})) \in \mathcal{R}^{2d+dk_H+2+m-|\mathcal{I}|}$ such that there exists some $\mathbf{h} \in \mathcal{R}^d$ with $\|\mathbf{h}\|_2 \leq B_h$ that verifies

$$\begin{aligned} & \mathbf{w}_1 + \mathbf{A}\mathbf{w}_2 + [q_L t \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3 - \sum_{i \in [m] \setminus \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\bar{\mathcal{I}})[i] - (q_\pi \bmod q)\mathbf{h} \\ &= \mathbf{u} + \sum_{i \in \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\mathcal{I})[i] \bmod q\mathcal{R} \\ & \|\mathbf{w}_i\|_2 \leq B_i + B_r + \sqrt{nd} =: B_{wi} \text{ for } i \in \{1, 2\}, \quad \|\mathbf{v}_3\|_2 \leq B_3 \\ & t \in \mathcal{T}_\omega, \text{ and } \forall i \in [m] \setminus \mathcal{I}, \mathbf{m}(\bar{\mathcal{I}})[i] \in T_1 \end{aligned}$$

The vectors $\mathbf{h}_{\text{iss},1}, \mathbf{h}_{\text{iss},2}, \mathbf{h}$ are computed from the extracted witness at the end of the soundness proofs of **(Prove_{iss}, Verify_{iss})** or **(Prove, Verify)**. As described in Section 4.2, the bounds $B_{h,\text{iss},i}, B_h$ depend on the lifting bounds, the approximate range proof bounds B_{arp} , and are divided by the proof modulus q_π . In our case, we have $q_\pi \bmod q = 1$ and $B_{h,\text{iss},i}, B_h$ are much smaller than the other terms they are added to, hence only entailing a negligible additive overhead.

6 Security Analysis

We now detail the security analysis of our anonymous credentials system. Both anonymity and unforgeability rely on the standard lattice assumptions M-LWE and M-SIS. We first prove the anonymity of our anonymous credentials system.

Theorem 6.1 (Anonymity). *The anonymous credentials of Section 5.2 is ε_{an} -anonymous with $\varepsilon_{an} = 2Q_{\text{iss}}(\varepsilon_{zk,\text{iss}} + \varepsilon_{\text{M-LWE}}^{(\text{com})}) + 2Q_{\text{iss}}Q_{\text{show}}\varepsilon_{zk}, \varepsilon_{zk,\text{iss}}, \varepsilon_{zk}$ being the respective zero-knowledge security of **(Prove_{iss}, Verify_{iss})** and **(Prove, Verify)** from Lemma A.1 and A.2, and $\varepsilon_{\text{M-LWE}}^{(\text{com})}$ the hardness bound of M-LWE _{$n,d,q,\mathcal{B}_{\text{pr}}^{2d}$} . The values Q_{iss} and Q_{show} represent the maximal numbers of issuance per organization key pair and showing per credential respectively.*

Proof. Our proof of anonymity proceeds with successive game hops to eventually make the entire view of the adversary independent of the bit b of the experiment of Figure 3.1. Note that our more permissive success conditions makes the whole reduction more challenging. Indeed, in the previous models where the revealed attributes of the challenge credentials (with index l_0 and l_1) must be equal, one must only ensure that the showing does not leak any information on the

other attributes or on the credential itself, which readily follows from the zero-knowledge property of π . Here, we allow the revealed attributes to differ under some natural conditions, which means that we must ensure that the adversary has no way to link these attributes to their credential. This adds constraints to the issuance procedure and all the previous showings of these credentials, and actually to all the credentials as the adversary may dynamically choose l_0 and l_1 . Concretely, we must ensure that the adversary's view is exactly $\text{State}_{\mathcal{A}}$, which will be done by simulating all the issuance and showing queries.

Let us now prove it formally. We simulate both experiments Game_{an-0} and Game_{an-1} simultaneously. To detail the changes, we call Game_0 the original experiment Game_{an-b} (regardless of the value of b) and $\text{Adv}_0[\mathcal{A}]$ the advantage of $\mathcal{A}$ in Game_0 , i.e., $\text{Adv}_0[\mathcal{A}] = \text{Adv}_{an}[\mathcal{A}]$. We then introduce games Game_i and define the advantage of $\mathcal{A}$, denoted by $\text{Adv}_i[\mathcal{A}]$, accordingly.

Game₁. We first simulate all the proofs π_{iss} in all the queries to $\mathcal{O}_{\text{Obtain}}$. Concretely, for each query, instead of generating the proof π_{iss} as $\pi_{\text{iss}} = \text{Prove}_{\text{iss}}(\text{statement}_{\text{iss}}, (\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}(\bar{\mathcal{T}})))$, it generates π_{iss} with the simulator of Lemma A.1 without resorting to the witness. Distinguishing between Game_0 and Game_1 then readily gives an adversary capable of distinguishing between at most Q_{iss} legitimate proofs and the same number of simulated ones, which would attack the zero-knowledge security of the issuance proof system. As a result, we directly get

$$|\text{Adv}_0[\mathcal{A}] - \text{Adv}_1[\mathcal{A}]| \leq 2Q_{\text{iss}}\varepsilon_{zk,\text{iss}},$$

where $\varepsilon_{zk,\text{iss}}$ is the zero-knowledge security of $(\text{Prove}_{\text{iss}}, \text{Verify}_{\text{iss}})$, and the factor 2 comes from the fact that we need to simulate them in both experiments ($b = 0$ and $b = 1$).

Game₂. We then simulate all the proofs π in all the queries to $\mathcal{O}_{\text{Show}}$ as well as that of the challenge showing. Just like in the previous game, we replace the legitimately generated proofs $\pi = \text{Prove}(\text{statement}, (\mathbf{w}_1, \mathbf{w}_2, \mathbf{v}_3, t, \mathbf{m}(\bar{\mathcal{T}})))$ by simulated ones. As there are at most Q_{show} showings of each credential, for each $l \in [Q_{\text{iss}}]$, the adversary may query $\mathcal{O}_{\text{Show}}(l, *)$ at most Q_{show} times. This bound can indeed be enforced by the challenger as the adversary gives out the issuance index as input. In the end, we essentially simulate $Q_{\text{iss}}Q_{\text{show}}$ proofs. Based on the zero-knowledge property of $(\text{Prove}, \text{Verify})$, we obtain

$$|\text{Adv}_1[\mathcal{A}] - \text{Adv}_2[\mathcal{A}]| \leq 2Q_{\text{iss}}Q_{\text{show}}\varepsilon_{zk},$$

where ε_{zk} is the zero-knowledge security of $(\text{Prove}, \text{Verify})$.

Game₃. We now simulate the commitments in each query to $\mathcal{O}_{\text{Obtain}}$. In Game_3 , instead of sampling $\mathbf{r}_1, \mathbf{r}_2$, the challenger samples $\mathbf{u}_c \leftarrow U(\mathcal{R}_q^d)$ and sets $\mathbf{c} = \mathbf{u}_c + \text{upk} + \mathbf{Dm} \bmod q\mathcal{R}$. Note this change is licit because the commitment randomness is no longer needed anywhere else than in generating the commitment. Indeed, in queries to $\mathcal{O}_{\text{Obtain}}$, the adversary never gets cred_{unb} (which would use $\mathbf{r}_2$) and only gets cred . Also, in queries to $\mathcal{O}_{\text{Show}}$, the zero-knowledge proofs π are simulated as of the previous game, and thus do not use cred_{unb} and in turn $(\mathbf{r}_1, \mathbf{r}_2)$.

A distinguisher between Game_2 and Game_3 can then be turned into a distinguisher for the multiple secret version of M-LWE $_{n,d,q,\mathcal{B}_{\eta}^{2d}}$, with at most Q_{iss}

secrets. Indeed, given the instance $(\mathbf{A}, \mathbf{u}_{c,1}, \dots, \mathbf{u}_{c,Q_{\text{iss}}})$, the reduction would simulate the challenger by using $\mathbf{A}$ in the public parameters, and $\mathbf{u}_{c,i}$ in the i -th query to $\mathcal{O}_{\text{Obtain}}$. If $\mathbf{u}_{c,i} = \mathbf{r}_{1,i} + \mathbf{A}\mathbf{r}_{2,i} \bmod q\mathcal{R}$ for $\mathbf{r}_{j,i} \sim \mathcal{B}_{\eta_r}^d$, the reduction perfectly simulates **Game**₂. If on the other hand $\mathbf{u}_{c,i} \sim U(\mathcal{R}_q^d)$, it simulates **Game**₃. We then have

$$|\text{Adv}_2[\mathcal{A}] - \text{Adv}_3[\mathcal{A}]| \leq 2Q_{\text{iss}}\varepsilon_{\text{M-LWE}}^{(\text{com})}$$

The factor Q_{iss} comes from the standard hybrid argument which reduces the multiple secret version from the single secret one. The term $\varepsilon_{\text{M-LWE}}^{(\text{com})}$ then corresponds to the hardness bound of the single secret version $\text{M-LWE}_{n,d,d,q,\mathcal{B}_{\eta_r}^{2d}}$.

Game₄. We finish the simulation of the commitments in the queries to $\mathcal{O}_{\text{Obtain}}$ by sampling the whole commitment uniformly. Hence, in **Game**₄, the challenger samples $\mathbf{c} \leftarrow U(\mathcal{R}_q^d)$. Because in **Game**₃ $\mathbf{u}_c$ was uniform and independent of $\text{upk} + \mathbf{Dm}$, the distribution of $\mathbf{c}$ in **Game**₃ was exactly the uniform distribution over $\mathcal{R}_q^d$. As a result, both games are identical, yielding

$$\text{Adv}_3[\mathcal{A}] = \text{Adv}_4[\mathcal{A}]$$

In **Game**₄, we are ensured that the view of the adversary is exactly $\text{State}_{\mathcal{A}}$ as all the attributes unknown to $\mathcal{A}$ have been simulated in the previous queries. If, $\forall i \in \mathcal{I}$, $\text{attr}_{l_0}[i] = \text{attr}_{l_1}[i]$, then we can readily conclude as the showing is completely independent of b . If $\exists i \in \mathcal{I}$ such that $\text{attr}_{l_0}[i] \neq \text{attr}_{l_1}[i]$, then the success condition ensures that these attributes are unknown to $\mathcal{A}$ and so that they were never involved in any previous oracle queries. Moreover, $\text{attr}_{l_0}[i]$ and $\text{attr}_{l_1}[i]$ have been selected by the reduction independently of all the other elements involved in the credentials, and therefore independently of b .

In both cases, this means that the adversary's view is independent of b and so that $\text{Adv}_4[\mathcal{A}] = 0$. With the prior inequalities and the triangle inequality, we then get $\text{Adv}_{\text{an}}[\mathcal{A}] \leq 2Q_{\text{iss}}(\varepsilon_{zk,\text{iss}} + \varepsilon_{\text{M-LWE}}^{(\text{com})}) + 2Q_{\text{iss}}Q_{\text{show}}\varepsilon_{zk}$ as claimed. $\square$

Theorem 6.2 (Unforgeability). *The anonymous credentials of Section 5.2 is ε_{uf} -unforgeable with*

$$\begin{aligned} \varepsilon_{uf} = \varepsilon_{\text{snd}} + |\mathcal{T}_w| & \left(Q_{\text{show}}\varepsilon_{zk} + \varepsilon_{zk,\text{iss}} + m\varepsilon_{\text{M-LWE}}^{(\text{pk})} + \frac{1 + \varepsilon}{1 - \varepsilon} \left(\varepsilon_{\text{snd},\text{iss}} \right. \right. \\ & \left. \left. + \frac{4M_1M_2}{(1-p)^3} \cdot \delta \cdot h^{\text{odk}_H} \left(\varepsilon_{\text{M-LWE}}^{(\text{pk})} + 2(Q_{\text{iss}} + 1) \left(\varepsilon_{\text{M-LWE}}^{(\text{upk})} + \varepsilon_{\text{M-SIS}} \right) \right) \right) \right), \end{aligned}$$

where $\mathcal{T}_w$ is the tag space, $\varepsilon_{\text{snd}}, \varepsilon_{zk}, \varepsilon_{\text{snd},\text{iss}}, \varepsilon_{zk,\text{iss}}$ are the soundness and zero-knowledge security of $(\text{Prove}_{\text{iss}}, \text{Verify}_{\text{iss}})$ and $(\text{Prove}, \text{Verify})$ from Lemmas A.1 and A.2, and $\varepsilon_{\text{M-LWE}}^{(\text{pk})}, \varepsilon_{\text{M-LWE}}^{(\text{upk})}, \varepsilon_{\text{M-SIS}}$ are the hardness bounds of $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1^{2d}}$, $\text{M-LWE}_{n,d,d,q,U(T_1^{2d})}$ and $\text{M-SIS}_{n,d,2d,q,\beta}$ respectively, for

$$\beta = \sqrt{(2B_{w1} + B_{h,\text{iss},1} + B_{h,\text{iss},2} + B_h)^2 + 4B_{w2}^2} + c\sqrt{nd}\sqrt{4B_3^2 + nm}.$$

Then, p is the tailcut probability with which the preimage bounds B_i are verified, and $h^{\circ dk_H}$ is the dk_H -th composition power of the function h defined by

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

$$\text{with } \mu = \left(1 + \frac{\lambda(2\lambda - 1)(\delta^2 - 1)^2}{(2 - \delta^2)^{2\lambda+1}} \right)^{(\text{Q}_{\text{iss}} - 1)/2\lambda}.$$

Finally, the term δ is the loss of **TSampler** from [JS26] which is equal to

$$\delta = \left(\frac{1 + \frac{\varepsilon}{dk_L}}{1 - \frac{\varepsilon}{dk_L}} \right)^{dk_L} \left(\frac{1 + \frac{\varepsilon}{dk_H}}{1 - \frac{\varepsilon}{dk_H}} \right)^{dk_H} \left(\frac{1 + \varepsilon}{1 - \varepsilon} \right)^{6d(n-1)+3} \approx 1 + (12d(n-1) + 10)\varepsilon.$$

The parameters h and δ are common for constructions based on **TSampler** from [JS26] (and previous versions). We refer to [AGJ⁺24] for a more detailed discussion on h but we simply point out that its impact on the bit security of the input is moderate.

Proof. We recall that the unforgeability experiment allows the adversary to win in two different ways. Either it manages to run the **Show** protocol with revealed attributes $\mathbf{m}(\mathcal{I})$ that are clearly inconsistent with those owned by corrupt users or it successfully produces $\text{ctr}_{\text{cor}} + 1$ inconsistent presentations, where ctr_{cor} is the number of certificates it owns. What we aim at proving is that, in both cases, the adversary must have used at some point a forged certificate or one that was issued to an honest user and hence bound to their secret key. Unfortunately, it is not always simple to decide when the adversary did that. More specifically, in the former case, the answer is straightforward as the fraudulent presentation can be identified from the revealed attributes. In the latter case, the answer is not that easy as all we know is that (at least) one of the $\text{ctr}_{\text{cor}} + 1$ presentations should not have happened. As we do not want to extract all the NIZK proofs, we will make a guess on the fraudulent presentation and only extract the corresponding NIZK proof. In all cases, this means that the probability to extract the right proof is at least $\frac{1}{\text{ctr}_{\text{cor}} + 1}$. Once this is done, we essentially fall back on the unforgeability of the signature scheme in [JS26].

Formally, we proceed through a series of game hops that only rely on statistical arguments or standard lattice assumptions. To detail the changes, we call **Game**₀ the original experiment **Game**_{*uf*} and $\text{Adv}_0[\mathcal{A}]$ the advantage of $\mathcal{A}$ in **Game**₀, i.e., $\text{Adv}_0[\mathcal{A}] = \text{Adv}_{uf}[\mathcal{A}]$. We then introduce games **Game**_{*i*} and define the advantage of $\mathcal{A}$, denoted by $\text{Adv}_i[\mathcal{A}]$, accordingly. To simplify notations, in this proof, we use $[p]_q$ to denote $p \bmod q$, where p is $q_{\pi, \text{iss}}$ (resp. q_{π}) is the proof modulus for **Issue** (resp. **Show**).

Game₁. A successful adversary $\mathcal{A}$ must have produced at least one valid presentation at Step 6 of the unforgeability experiment. If this presentation fulfils the condition in Step 8 of the same experiment, then we extract the proof π^* it contains. If no presentation satisfies this condition, then the adversary can only win

by producing $\text{ctr}_{\text{cor}} + 1$ inconsistent presentations. We then select a random k in $[1, \text{ctr}_{\text{cor}} + 1]$ and extract the proof π^* included in the k -th presentation. In both cases, unless the adversary broke soundness, the challenger is able to extract $(\mathbf{w}_1^*, \mathbf{w}_2^*, \mathbf{v}_3^*, t^*, \mathbf{m}(\bar{\mathcal{I}})^*, \mathbf{h}^*)$ with the extractor of Lemma A.2, which satisfies

$$\begin{aligned} & \mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + [q_L t^* \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3^* - \sum_{i \in [m] \setminus \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\bar{\mathcal{I}})^*[i] - [q_\pi]_q \mathbf{h}^* \\ &= \mathbf{u} + \sum_{i \in \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\mathcal{I})^*[i] \bmod q\mathcal{R} \\ & \|\mathbf{w}_i^*\|_2 \leq B_i + B_r + \sqrt{nd} =: B_{wi} \text{ for } i \in \{1, 2\}, \quad \|\mathbf{v}_3^*\|_2 \leq B_3, \quad \|\mathbf{h}^*\|_2 \leq B_h \\ & t^* \in \mathcal{T}_\omega, \text{ and } \forall i \in [m] \setminus \mathcal{I}, \mathbf{m}(\bar{\mathcal{I}})^*[i] \in T_1 \end{aligned}$$

We write $\mathbf{m}^*$ the vector such that $\forall S \in \{\mathcal{I}, \bar{\mathcal{I}}\}, \forall i \in S, \mathbf{m}^*[i] = \mathbf{m}(S)^*[i]$. Said differently, we complete the $\emptyset$ entries of $\mathbf{m}(\mathcal{I})^*$ with the extracted attributes, i.e., $\mathbf{m}^*$ is the unique attribute vector such that $\mathbf{m}^*(\mathcal{I}) = \mathbf{m}(\mathcal{I})^*$ and $\mathbf{m}^*(\bar{\mathcal{I}}) = \mathbf{m}(\bar{\mathcal{I}})^*$. By Lemma A.2, we then have

$$\text{Adv}_0[\mathcal{A}] \leq \text{Adv}_1[\mathcal{A}] + \varepsilon_{\text{snd}},$$

where ε_{snd} is the soundness error of (Prove, Verify) from Lemma A.2.

Game₂. We now introduce a guess on the tag t^* , but somewhat indirectly as we sample $l^+ \leftarrow U(|\mathcal{T}_w|)$ and define $t^+ = F(\text{st}_0 + l^+)$. When extracting π^* as in Game₁, the challenger now checks whether the guess was correct and aborts if $t^* \neq t^+$. Because l^+ (and as such t^+) is uniform and independent from the view of $\mathcal{A}$, it directly holds that

$$\text{Adv}_1[\mathcal{A}] \leq |\mathcal{T}_w| \cdot \text{Adv}_2[\mathcal{A}],$$

Game₃. The way t^+ is generated from l^+ implies that t^+ will be used during the l^+ -th query and most of the following games will describe how the challenger should behave to handle this query. However, the adversary may make less than l^+ queries, in which case all the elements depending on this specific query would be undefined. To avoid forking our security proof to tackle these two cases separately, we will simply force the challenger to locally generate the credential as if it had received a l^+ -th query from the adversary. This way, all the elements associated with the l^+ -th query will be properly defined in the remainder of this proof. By locally, we mean that the adversary receives no information on this credential, which ensures that its behaviour is not affected.

Concretely, if the experiment stops before the adversary makes the l^+ query, i.e., with $\text{ctr} < l^+$, the challenger proceeds as follows. If $\text{HU} \neq \emptyset$, it samples $j \leftarrow U(\text{HU})$ and locally performs the query $\mathcal{O}_{\text{Oblss}}(j, \mathbf{m}^*)$ (where $\mathbf{m}^*$ is the attribute vector extracted from the selected Show in Game₁) as if it was the l^+ -th query (in particular, it uses t^+). If $\text{HU} = \emptyset$, it locally creates an honest user and proceeds as previously.

As this extra local query is independent of the view of $\mathcal{A}$, both games are identically distributed. Thence, we have

$$\text{Adv}_2[\mathcal{A}] = \text{Adv}_3[\mathcal{A}].$$

Game₄. We now proceed by simulating certain zero-knowledge proofs. Depending on whether the l^+ -th issuance corresponds in the end to a credential of an honest user or not, we may need to ensure that this honest user's secret key did not leak during the issuance or subsequent showings. We preemptively make the following change. For each query $\mathcal{O}_{\text{Show}}(l^+, *)$, the challenger simulates the zero-knowledge proof π by using the simulator of Lemma A.2. Additionally, if the l^+ -th issuance corresponds to a query to $\mathcal{O}_{\text{ObtIss}}$ (as opposed to $\mathcal{O}_{\text{Issue}}$), the challenger simulates the proof π_{iss} when running **Issue** in this query, by calling the simulator of Lemma A.1. Indeed, in $\mathcal{O}_{\text{ObtIss}}$, the challenger generates a credential to an honest user and thus assumes the roles of both the organization and the user. In that case, it can simulate the issuance proof π_{iss} .

In the end, the challenger simulates at most Q_{Show} proofs in **Show** in the queries to $\mathcal{O}_{\text{Show}}$, and at most one proof in **Issue**. We then have

$$\text{Adv}_3[\mathcal{A}] \leq \text{Adv}_4[\mathcal{A}] + Q_{\text{Show}}\varepsilon_{zk} + \varepsilon_{zk, \text{iss}},$$

where $\varepsilon_{zk, \text{iss}}$ and ε_{zk} are the zero-knowledge security bound of (**Prove_{iss}**, **Verify_{iss}**) and (**Prove**, **Verify**) respectively, which are defined in Lemmas A.1 and A.2.

Game₅. We now simulate the commitment matrix $\mathbf{D}$ by embedding a hidden short relation. More precisely, the challenger samples $\mathbf{S}_1, \mathbf{S}_2$ from $\mathcal{B}_1^{d \times m}$ and sets $\mathbf{D} = \mathbf{S}_1 + \mathbf{A}\mathbf{S}_2 \bmod q\mathcal{R}$. It then programs $\mathcal{H}(\text{seed}, \text{dsep}_4)$ to output $\mathbf{D}$. This correctly simulates $\mathbf{D}$ under the $\text{M-LWE}_{n, d, d, q, \mathcal{B}_1^{2d}}^m$ assumption, which is the variant with m independent (secret, error) vectors.

Indeed, let $\mathcal{D}$ be a distinguisher of the views of $\mathcal{A}$ between **Game₄** and **Game₅**. We construct a distinguisher $\mathcal{B}$ for M-LWE as follows. On input $(\mathbf{A}, \mathbf{D}) \in \mathcal{R}_q^{d \times d} \times \mathcal{R}_q^{d \times m}$, the reduction programs $\mathcal{H}(\text{seed}, \text{dsep}_1) = \mathbf{A}$ and $\mathcal{H}(\text{seed}, \text{dsep}_4) = \mathbf{D}$. It then proceeds as usual and calls $\mathcal{D}$ on the resulting view of $\mathcal{A}$. If $\mathcal{D}$ returns 0 (for **Game₄**), $\mathcal{B}$ returns 0 (for uniform). Otherwise, it returns 1 (for M-LWE). Let us now verify the reduction is correct. First, because $\mathbf{A}$ is uniform in $\mathcal{R}_q^{d \times d}$, it is perfectly simulated in the ROM. Then, if $\mathbf{D}$ is uniform, this perfectly simulates **Game₄**. On the other hand, if $\mathbf{D} = \mathbf{S}_1 + \mathbf{A}\mathbf{S}_2 \bmod q\mathcal{R}$ for $\mathbf{S}_i \leftarrow \mathcal{B}_1^{d \times m}$, this perfectly simulates **Game₅**. As a result, if $\varepsilon_{\text{M-LWE}^m}^{(\text{pk})}$ denotes the hardness bound of the m -secrets version, we have

$$\text{Adv}_4[\mathcal{A}] \leq \text{Adv}_5[\mathcal{A}] + \varepsilon_{\text{M-LWE}^m}^{(\text{pk})} \leq \text{Adv}_5[\mathcal{A}] + m\varepsilon_{\text{M-LWE}}^{(\text{pk})}.$$

The last inequality comes from the standard hybrid argument showing a reduction from the single-secret version to the multiple secrets one, as long as m is polynomial, which is our case. Here, $\varepsilon_{\text{M-LWE}}^{(\text{pk})}$ denotes the hardness bound of $\text{M-LWE}_{n, d, d, q, \mathcal{B}_1^{2d}}$.

Game₆. We now start changing the generation of the public syndrome $\mathbf{u}$ so as to hide a preimage within it. This preimage later serves in a single query where the trapdoor vanishes. The challenger samples $v_{3,2} \leftarrow \mathcal{D}_{\mathcal{R}, s_3}$ and $\mathbf{u}' \leftarrow U(\mathcal{R}_q^d)$, and sets $\mathbf{u} = \mathbf{u}' + \mathbf{a} \cdot v_{3,2} \bmod q\mathcal{R}$. It then reprograms $\mathcal{H}(\text{seed}, \text{dsep}_2) = \mathbf{u}$. Because $\mathbf{u}'$ is uniform and independent of $\mathbf{a} \cdot v_{3,2}$, it holds that $\mathbf{u}$ is exactly uniform as desired. Hence, both games are identically distributed and

$$\text{Adv}_5[\mathcal{A}] = \text{Adv}_6[\mathcal{A}].$$

Game₇. We now hide the rest of the preimage within $\mathbf{u}'$. The challenger samples $\mathbf{v}_1 \leftarrow \mathcal{D}_{\mathcal{R}^d, s_1}$, $\mathbf{v}_2 \leftarrow \mathcal{D}_{\mathcal{R}^d, s_2}$ and $\mathbf{v}_{3,1} \leftarrow \mathcal{D}_{\mathcal{R}^{dk_H}, s_3}$ and sets $\mathbf{u}' = \mathbf{v}_1 + \mathbf{A}\mathbf{v}_2 + [q_L t^+ \mathbf{G}_H - \mathbf{B}]\mathbf{v}_{3,1} \bmod q\mathcal{R}$. It then defines $\mathbf{u} = \mathbf{u}' + \mathbf{a} \cdot v_{3,2} \bmod q\mathcal{R}$ as before. Defining $\mathbf{v}_3 = [\mathbf{v}_{3,1}^T | v_{3,2}]^T$ yields $\mathbf{u} = \mathbf{v}_1 + \mathbf{A}\mathbf{v}_2 + [q_L t^+ \mathbf{G}_H - \mathbf{B}]\mathbf{a} \mathbf{v}_3 \bmod q\mathcal{R}$. Because of how the Gaussian parameters s_1, s_2, s_3 of **TSampler** are set, we have that $\sqrt{\text{diag}(s_1^2 \mathbf{I}_{nd}, s_2^2 \mathbf{I}_{nd}, s_3^2 \mathbf{I}_{ndk_H})} \geq \eta_\varepsilon(\mathcal{L}_q^\perp([\mathbf{I}_d | \mathbf{A}] [q_L t^+ \mathbf{G}_H - \mathbf{B}]))$. The regularity lemma of Lemma 2.1 then gives

$$\text{Adv}_6[\mathcal{A}] = \frac{1 + \varepsilon}{1 - \varepsilon} \text{Adv}_7[\mathcal{A}].$$

Game₈. We now extract the issuance proof π_{iss} used in the l^+ -th issuance if it so happens that this issuance index corresponds to a query to $\mathcal{O}_{\text{Issue}}(j, \mathbf{m}(\mathcal{I}'))$. As the proof must be valid, unless the adversary broke soundness, the challenger is able to extract $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}(\overline{\mathcal{I}'}), \mathbf{h}_{\text{iss},1}, \mathbf{h}_{\text{iss},2})$ with the extractor of Lemma A.1, which satisfies

$$\begin{aligned} \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \sum_{i \in [m] \setminus \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\overline{\mathcal{I}'})[i] - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},1} \\ = \mathbf{c} - \text{upk}_j - \sum_{i \in \mathcal{I}} \mathbf{d}_i \cdot \mathbf{m}(\mathcal{I}')[i] \bmod q\mathcal{R} \\ \mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},2} = \text{upk}_j \bmod q\mathcal{R}, \quad \|\mathbf{r}_i\|_2 \leq B_r \text{ for } i \in \{1, 2\}, \\ \mathbf{s}_i \in T_1^d \text{ for } i \in \{1, 2\}, \text{ and } \forall i \in [m] \setminus \mathcal{I}', \mathbf{m}(\overline{\mathcal{I}'})[i] \in T_1 \\ \|\mathbf{h}_{\text{iss},i}\|_2 \leq B_{h, \text{iss},i} \text{ for } i \in \{1, 2\}. \end{aligned}$$

From the extracted values, we complete $\mathbf{m}(\mathcal{I}')$ and define $\mathbf{m}$ as the vector such that $\forall S \in \{\mathcal{I}', \overline{\mathcal{I}'}\}, \forall i \in S, \mathbf{m}[i] = \mathbf{m}(S)[i]$. We then have $\mathbf{c} = \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \text{upk}_j + \mathbf{D}\mathbf{m} - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},1} \bmod q\mathcal{R}$. By Lemma A.1, it holds that

$$\text{Adv}_7[\mathcal{A}] \leq \text{Adv}_8[\mathcal{A}] + \varepsilon_{\text{snd}, \text{iss}},$$

where $\varepsilon_{\text{snd}, \text{iss}}$ is the soundness error of $(\text{Prove}_{\text{iss}}, \text{Verify}_{\text{iss}})$ from Lemma A.1.

We note that if the l^+ -th issuance happens instead in a query $\mathcal{O}_{\text{ObtIss}}(j, \mathbf{m})$ or is dealt with locally as in **Game₃**, the challenger knows $\mathbf{s}_1, \mathbf{s}_2$ underlying upk_j as j must be in **HU**. It also selects $\mathbf{r}_1, \mathbf{r}_2$ by itself, and the oracle is queried with the entire attribute vector $\mathbf{m}$. In that case, no extraction is needed, in particular because the challenger generates π_{iss} itself, and actually simulates it as of **Game₄**.

As the proof is not extracted, there is no longer additive slacks $\mathbf{h}_{\text{iss},i}$ in the equations above. Said differently, in this case we would have $\mathbf{h}_{\text{iss},i} = \mathbf{0}$. To unify further arguments, we keep the most general case where there is a slack $\mathbf{h}_{\text{iss},i}$, which is possibly non-zero. Regardless, either way, the challenger knows some $\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}$ satisfying the correct equations for the l^+ -th issuance. Note that we could merge $\mathbf{h}_{\text{iss},1}$ with $\mathbf{r}_1$ and $\mathbf{h}_{\text{iss},2}$ with $\mathbf{s}_1$ which could then be treated later on as if there were no slack.

Game₉. Before making the trapdoor vanish to embed our guess t^+ within $\mathbf{B}$, we need to modify the response of the l^+ -th issuance query so as to still be able to produce a credential. For that, we want to use the preimage $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ embedded in $\mathbf{u}$, but it does not yet account for the commitment. Using the short relations we introduced and the now known elements $\mathbf{r}_1, \mathbf{r}_2, \mathbf{s}_1, \mathbf{s}_2, \mathbf{m}, \mathbf{h}_{\text{iss},1}, \mathbf{h}_{\text{iss},2}$, we can tweak the response so as to ensure the correct equation is verified. This tweak however skews the distribution of legitimate preimages. We then resort to rejection sampling to ensure the preimage tweak goes unnoticed for the adversary. We thus prepare the rejection sampling progressively.

For the l^+ -th issuance query, the challenger samples $u_1, u_2 \leftarrow U([0, 1])$ and aborts the game if one the following booleans is true:

$$\begin{aligned} & \llbracket u_1 > 1/M_1 \rrbracket, \quad \llbracket u_2 > 1/M_2 \rrbracket, \\ & \llbracket \langle \tau(\mathbf{v}_1), \tau(\mathbf{r}_1 + \mathbf{s}_1 + \mathbf{S}_1 \mathbf{m} - [q_{\pi, \text{iss}}]_q(\mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2})) \rangle < 0 \rrbracket, \\ & \llbracket \langle \tau(\mathbf{v}_2), \tau(\mathbf{r}_2 + \mathbf{s}_2 + \mathbf{S}_2 \mathbf{m}) \rangle < 0 \rrbracket. \end{aligned}$$

If all false, the challenger continues the game as before. Because $\mathbf{v}_1, \mathbf{v}_2$ are centered, the last two booleans are true with probability $1/2$. It then holds that the challenger continues with probability $1/(4M_1M_2)$. If so, the view of $\mathcal{A}$ is unchanged and therefore

$$\text{Adv}_8[\mathcal{A}] \leq 4M_1M_2 \cdot \text{Adv}_9[\mathcal{A}],$$

Game₁₀. We now tweak the values $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ so that they satisfy the correct equation. But at this stage we keep generating the credential using **TSampler** regardless. Concretely, the challenger defines $\mathbf{v}_1^+ = \mathbf{v}_1 + \mathbf{r}_1 + \mathbf{s}_1 + \mathbf{S}_1 \mathbf{m} - [q_{\pi, \text{iss}}]_q(\mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2})$, $\mathbf{v}_2^+ = \mathbf{v}_2 + \mathbf{r}_2 + \mathbf{s}_2 + \mathbf{S}_2 \mathbf{m}$, and $\mathbf{v}_3^+ = \mathbf{v}_3$. It samples $u_1, u_2 \leftarrow U([0, 1])$ and computes $\delta_1 = \langle \tau(\mathbf{v}_1^+), \tau(\mathbf{r}_1 + \mathbf{s}_1 + \mathbf{S}_1 \mathbf{m} - [q_{\pi, \text{iss}}]_q(\mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2})) \rangle$ and $\delta_2 = \langle \tau(\mathbf{v}_2^+), \tau(\mathbf{r}_2 + \mathbf{s}_2 + \mathbf{S}_2 \mathbf{m}) \rangle$. Then, the challenger aborts if one of the following booleans is true:

$$\begin{aligned} & \llbracket u_1 > \exp(\pi(\|\mathbf{r}_1 + \mathbf{s}_1 + \mathbf{S}_1 \mathbf{m} - [q_{\pi, \text{iss}}]_q(\mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2})\|_2^2 - 2\delta_1)/s_1^2)/M_1 \rrbracket, \\ & \llbracket u_2 > \exp(\pi(\|\mathbf{r}_2 + \mathbf{s}_2 + \mathbf{S}_2 \mathbf{m}\|_2^2 - 2\delta_2)/s_2^2)/M_2 \rrbracket, \\ & \llbracket \delta_1 < 0 \rrbracket, \quad \llbracket \delta_2 < 0 \rrbracket. \end{aligned}$$

Based on the parameters $s_1, s_2, \alpha_1, \alpha_2, M_1, M_2$, when the challenger does not abort, the game remains identically distributed by Lemma 2.3. In particular, we note that, for now, the $\mathbf{v}_i^+$ are never sent to $\mathcal{A}$. The response to the l^+ -th

issuance is still sampled using **TSampler**. Hence,

$$\text{Adv}_9[\mathcal{A}] = \text{Adv}_{10}[\mathcal{A}].$$

Game₁₁. Before being able to use the $\mathbf{v}_i^+$ in the l^+ -th query, we must ensure they verify the correct norm bound. After rejection sampling, the challenger aborts if $\llbracket \|\mathbf{v}_1^+\|_2 > B_1 \rrbracket$ or $\llbracket \|\mathbf{v}_2^+\|_2 > B_2 \rrbracket$ or $\llbracket \|\mathbf{v}_3^+\|_2 > B_3 \rrbracket$. If on the contrary, they all pass the bounds, the challenger continues. By the rejection sampling result, it holds that $\mathbf{v}_i^+$ is Gaussian with parameter s_i . Using the tail bound of Lemma 2.2, it then holds that $\mathbb{P}[\forall i \in [3], \|\mathbf{v}_i^+\|_2 \leq B_i] \geq (1-p)^3$, where p is the tailcut probability for a single tail bound. Hence, we get

$$\text{Adv}_{10}[\mathcal{A}] \leq \frac{1}{(1-p)^3} \text{Adv}_{11}[\mathcal{A}].$$

Game₁₂. We now make the switch and use $(\mathbf{v}_1^+, \mathbf{v}_2^+, \mathbf{v}_3^+)$ to answer the l^+ -th issuance query. Hence, instead of using **TSampler**, the challenger simply sends $(t^+, \mathbf{v}_2^+, \mathbf{v}_3^+)$ in step 12 of **Issue**. Let us now analyze the distribution. As the tail bound truncation is applied in both games, the data processing inequality allows us to study the untruncated distributions instead.

It holds that $\mathbf{v}_i^+$ (before truncation) follows a Gaussian with parameter s_i . Additionally, we have

$$\begin{aligned} & \mathbf{v}_1^+ + \mathbf{A}\mathbf{v}_2^+ + [q_L t^+ \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3^+ \\ &= \mathbf{u} + (\mathbf{r}_1 + \mathbf{s}_1 + \mathbf{S}_1 \mathbf{m} - [q_{\pi, \text{iss}}]_q (\mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2})) + \mathbf{A}(\mathbf{r}_2 + \mathbf{s}_2 + \mathbf{S}_2 \mathbf{m}) \bmod q\mathcal{R} \\ &= \mathbf{u} + \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + (\mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},2}) \\ & \quad + (\mathbf{S}_1 + \mathbf{A}\mathbf{S}_2) \mathbf{m} - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},1} \bmod q\mathcal{R} \\ &= \mathbf{u} + \mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \text{upk}_j + \mathbf{D}\mathbf{m} - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},1} \bmod q\mathcal{R} \\ &= \mathbf{u} + \mathbf{c} \bmod q\mathcal{R}. \end{aligned}$$

The third equality holds because $\text{upk}_j = \mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 - [q_{\pi, \text{iss}}]_q \mathbf{h}_{\text{iss},2} \bmod q\mathcal{R}$ by the extraction of **Game₈** (with $\mathbf{h}_{\text{iss},2} = \mathbf{0}$ if the l^+ -th issuance corresponds to a query to $\mathcal{O}_{\text{Oblss}}$), and because $\mathbf{D} = \mathbf{S}_1 + \mathbf{A}\mathbf{S}_2 \bmod q\mathcal{R}$. The last equality holds also by the extraction of **Game₈** (with $\mathbf{h}_{\text{iss},1} = \mathbf{0}$ if the l^+ -th issuance corresponds to a query to $\mathcal{O}_{\text{Oblss}}$). As a result, $\mathbf{v}^+ = [\mathbf{v}_1^{+T} | \mathbf{v}_2^{+T} | \mathbf{v}_3^{+T}]^T$ follows the distribution $\mathcal{D}_{\mathcal{R}^{2d+dk_H+1}, \sqrt{\mathbf{S}}}$ conditioned on $[\mathbf{I}_d | \mathbf{A} | q_L t^+ \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}^+ = \mathbf{u} + \mathbf{c} \bmod q\mathcal{R}$, where $\mathbf{S} = \text{diag}(s_1^2 \mathbf{I}_{nd}, s_2^2 \mathbf{I}_{nd}, s_3^2 \mathbf{I}_{n(dk_H+1)})$. This conditional distribution exactly matches $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}+\mathbf{c}}([\mathbf{I}_d | \mathbf{A} | q_L t^+ \mathbf{G}_H - \mathbf{B}|\mathbf{a}], \sqrt{\mathbf{S}})}$, which is the ideal distribution targetted by **TSampler**. By the security of **TSampler** proven in [JS26, Thm. 3.1 & Cor. 3.1], it holds that the views of $\mathcal{A}$ are δ -close. More precisely, we have

$$\text{Adv}_{11}[\mathcal{A}] \leq \delta \cdot \text{Adv}_{12}[\mathcal{A}],$$

where

$$\delta = \left(\frac{1 + \frac{\varepsilon}{dk_L}}{1 - \frac{\varepsilon}{dk_L}} \right)^{dk_L} \left(\frac{1 + \frac{\varepsilon}{dk_H}}{1 - \frac{\varepsilon}{dk_H}} \right)^{dk_H} \left(\frac{1 + \varepsilon}{1 - \varepsilon} \right)^{6d(n-1)+3} \approx 1 + (12d(n-1) + 10)\varepsilon.$$

Game₁₃. Now that we removed the need for a trapdoor for the l^+ -th query featuring the tag guess t^+ , we can proceed with tweaking the organization's public key $\mathbf{B}$ so as to cancel the gadget matrix $\mathbf{G}_H$ for this specific tag. This step is crucial for extracting a valid M-SIS solution at the end of the proof. For that, we carry the exact trapdoor switching hybrid argument from [JS26]. To avoid repetition with the latter work, we compact their whole hybrid argument into one game hop and detail the complete loss at once. In Game₁₃, the challenger changes the construction of $\mathbf{B}$ by instead sampling $\mathbf{R}_1, \mathbf{R}_2 \leftarrow \mathcal{B}_1^{d \times dk_H}$ such that $\|\mathbf{R}_i\|_2 \leq B_R$ and defining $\mathbf{B} = \mathbf{R}_1 + \mathbf{A}\mathbf{R}_2 + q_L t^+ \mathbf{G}_H \bmod q\mathcal{R}$. Using the hybrid argument from [JS26, full version, Thm. 4.1], we obtain

$$\text{Adv}_{12}[\mathcal{A}] \leq h^{\circ dk_H}(\text{Adv}_{13}[\mathcal{A}]),$$

where $h^{\circ dk_H}$ is the dk_H -th composition power of the function h defined by

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

and where $\varepsilon_{\text{M-LWE}}^{(\text{pk})}$ is the hardness bound of $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1^{2d}}$, and

$$\mu = \left(1 + \frac{\lambda(2\lambda - 1)(\delta^2 - 1)^2}{(2 - \delta^2)^{2\lambda+1}} \right)^{(\text{Q}_{\text{iss}} - 1)/2\lambda}.$$

The details regarding the overall loss and how the function h affect the bit-security can be found in [JS26, full version]. The trapdoor switching argument relies on the column $\mathbf{a}$ to hide a partial trapdoor to still be able to produce credentials. The latter can be changed at each game hops by reprogramming $\mathcal{H}(\text{seed}, \text{dsep}_3)$.

Game₁₄. We further simulate the matrix by hiding a short relation within $\mathbf{a}$. The challenger samples $\mathbf{z}_1, \mathbf{z}_2 \leftarrow \mathcal{B}_1^d$ and defines $\mathbf{a} = \mathbf{z}_1 + \mathbf{A}\mathbf{z}_2 \bmod q\mathcal{R}$. It then reprograms $\mathcal{H}(\text{seed}, \text{dsep}_3) = \mathbf{a}$. Using the same argument as in Game₄ but in the single secret case, we get

$$\text{Adv}_{13}[\mathcal{A}] \leq \text{Adv}_{14}[\mathcal{A}] + \varepsilon_{\text{M-LWE}}^{(\text{pk})}.$$

Game₁₅. We now enforce a structured Johnson-Lindenstrauss bound on elements recovered from the forgery $\mathbf{v}_3^*, \mathbf{m}^*$, and the elements $\mathbf{v}_3^+, \mathbf{m}$ used in the l^+ -th query (which always exists as of Game₃). Namely, the challenger aborts at the end of the game if

$$\left\| \begin{bmatrix} \mathbf{R}_1 - \mathbf{z}_1 & \mathbf{S}_1 \\ \mathbf{R}_2 - \mathbf{z}_2 & \mathbf{S}_2 \end{bmatrix} \Delta_{3m} \right\|_2 > c\sqrt{nd} \|\Delta_{3m}\|_2, \quad \text{with } \Delta_{3m} = \begin{bmatrix} \mathbf{v}_3^* - \mathbf{v}_3^+ \\ \mathbf{m}^* - \mathbf{m} \end{bmatrix}$$

By the concentration bound on $\|\cdot\|_{2,\mathbb{Z}}$ of a structured binomial matrix from [AGJ⁺24, JS25a] and used in most prior works, the bound is satisfied with probability at least 1/2 for c very close to 1. It was empirically verified in [DFJS26] that $c \approx 1.061$. Hence, we have

$$\text{Adv}_{14}[\mathcal{A}] \leq 2 \cdot \text{Adv}_{15}[\mathcal{A}].$$

Reduction from M-SIS. We now exploit a successful adversary winning Game_{15} to solve M-SIS $_{n,d,2d,q,\beta}$. The reduction receives the M-SIS instance $[\mathbf{I}_d|\mathbf{A}]$ and is asked to produce $\mathbf{x}^* \neq \mathbf{0}$ such that $\|\mathbf{x}^*\|_2 \leq \beta$ and $[\mathbf{I}_d|\mathbf{A}]\mathbf{x}^* = \mathbf{0} \bmod q\mathcal{R}$. The reduction programs $\mathcal{H}(\text{seed}, \text{dsep}_1) = \mathbf{A}$ and proceeds exactly as in Game_{15} . Then, by Game_1 and Game_2 , we have

$$\mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + [-\mathbf{R}_1 - \mathbf{A}\mathbf{R}_2|\mathbf{z}_1 + \mathbf{A}\mathbf{z}_2]\mathbf{v}_3^* - [q_\pi]_q \mathbf{h}^* = \mathbf{u} + (\mathbf{S}_1 + \mathbf{A}\mathbf{S}_2)\mathbf{m}^*,$$

which can be re-written as

$$\left(\mathbf{w}_1^* - [\mathbf{R}_1| - \mathbf{z}_1|\mathbf{S}_1] \begin{bmatrix} \mathbf{v}_3^* \\ \mathbf{m}^* \end{bmatrix} - [q_\pi]_q \mathbf{h}^* \right) + \mathbf{A} \left(\mathbf{w}_2^* - [\mathbf{R}_2| - \mathbf{z}_2|\mathbf{S}_2] \begin{bmatrix} \mathbf{v}_3^* \\ \mathbf{m}^* \end{bmatrix} \right) = \mathbf{u},$$

From the l^+ -th query, we also have, for $\mathbf{h}_{\text{iss}} = \mathbf{h}_{\text{iss},1} + \mathbf{h}_{\text{iss},2}$ for short, that

$$\begin{aligned} \mathbf{u} = & \left(\mathbf{v}_1^+ - \mathbf{r}_1 - \mathbf{s}_1 + [q_{\pi,\text{iss}}]_q \mathbf{h}_{\text{iss}} - [\mathbf{R}_1| - \mathbf{z}_1|\mathbf{S}_1] \begin{bmatrix} \mathbf{v}_3^+ \\ \mathbf{m} \end{bmatrix} \right) \\ & + \mathbf{A} \left(\mathbf{v}_2^+ - \mathbf{r}_2 - \mathbf{s}_2[\mathbf{R}_2| - \mathbf{z}_2|\mathbf{S}_2] \begin{bmatrix} \mathbf{v}_3^+ \\ \mathbf{m} \end{bmatrix} \right), \end{aligned}$$

Defining

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{w}_1^* - \mathbf{v}_1^+ + \mathbf{r}_1 + \mathbf{s}_1 - [q_{\pi,\text{iss}}]_q \mathbf{h}_{\text{iss}} - [q_\pi]_q \mathbf{h}^* \\ \mathbf{w}_2^* - \mathbf{v}_2^+ + \mathbf{r}_2 + \mathbf{s}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{R}_1 - \mathbf{z}_1 & \mathbf{S}_1 \\ \mathbf{R}_2 - \mathbf{z}_2 & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_3^* - \mathbf{v}_3^+ \\ \mathbf{m}^* - \mathbf{m} \end{bmatrix},$$

then yields $[\mathbf{I}_d|\mathbf{A}]\mathbf{x}^* = \mathbf{0} \bmod q\mathcal{R}$. It remains to show that $\|\mathbf{x}^*\|_2 \leq \beta$ and $\mathbf{x}^* \neq \mathbf{0}$. We let $\Delta_1 = \mathbf{w}_1^* - \mathbf{v}_1^+ + \mathbf{r}_1 + \mathbf{s}_1 - [q_{\pi,\text{iss}}]_q \mathbf{h}_{\text{iss}} - [q_\pi]_q \mathbf{h}^*$ and $\Delta_2 = \mathbf{w}_2^* - \mathbf{v}_2^+ + \mathbf{r}_2 + \mathbf{s}_2$. First, by the bounds proven on $\mathbf{w}_i^*$, the truncation in Game_{11} for $\mathbf{v}_i^+$, the bounds enforced at issuance on $\mathbf{r}_i, \mathbf{s}_i$, and the extraction bound on $\mathbf{h}_{\text{iss}}, \mathbf{h}^*$ (note that the bound is valid even if $\mathbf{h}_{\text{iss},i} = \mathbf{0}$), we get $\|\Delta_1\|_2 \leq B_{w1} + B_1 + B_r + \sqrt{nd} + [q_{\pi,\text{iss}}]_q(B_{h,\text{iss},1} + B_{h,\text{iss},2}) + [q_\pi]_q B_h = 2B_{w1} + [q_{\pi,\text{iss}}]_q(B_{h,\text{iss},1} + B_{h,\text{iss},2}) + [q_\pi]_q B_h$. Similarly, we have $\|\Delta_2\|_2 \leq 2B_{w2}$. We also have $\|\Delta_{3m}\|_2 \leq \sqrt{4B_3^2 + nm}$. Because in practice we choose $[q_{\pi,\text{iss}}]_q = [q_\pi]_q = 1$, we omit them to simplify the bounds. A different choice would need to account for these factors in the final M-SIS bound. In the end, by the bound enforced in Game_{15} , we get

$$\|\mathbf{x}^*\|_2 \leq \sqrt{(2B_{w1} + B_{h,\text{iss},1} + B_{h,\text{iss},2} + B_h)^2 + 4B_{w2}^2} + c\sqrt{nd}\sqrt{4B_3^2 + nm} = \beta$$

To show $\mathbf{x}^* \neq \mathbf{0}$, we need to distinguish several cases. Let $(j, l^+, \text{cred}_{l^+}, \text{attr}_{l^+})$ be the entry of State_C for the l^+ -th issuance. By Game_3 , it always exists.

(1) If the credential associated to t^+ belongs to an honest user, i.e., $j \in \text{HU}$, (either because the l^+ -th issuance was made through a query to $\mathcal{O}_{\text{Oblss}}$ or because the adversary made less than l^+ query, as detailed in Game_3), then we rely on the unpredictability of $(\mathbf{s}_1, \mathbf{s}_2)$. Indeed, as we ensured the view of the adversary did not depend on $(\mathbf{s}_1, \mathbf{s}_2)$ but only on upk_j (in particular with the simulation of zero-knowledge proofs in Game_4), an adversary managing to generate elements so that $\mathbf{x}^* = \mathbf{0}$ could readily be used to solve search M-LWE $_{n,d,d,q,U(T_1^{2d})}$. In that

case, we deduce $\text{Adv}_{15}[\mathcal{A}] \leq \varepsilon_{\text{M-LWE}}^{(\text{upk})} + \varepsilon_{\text{M-SIS}}$, where $\varepsilon_{\text{M-LWE}}^{(\text{upk})}$ is the hardness bound of search M-LWE $_{n,d,d,q,U(T_1^{2d})}$, and $\varepsilon_{\text{M-SIS}}$ is that of M-SIS $_{n,d,2d,q,\beta}$. Note that in this case $\mathbf{h}_{\text{iss},i} = \mathbf{0}$, which could yield a slightly tighter M-SIS bound.

(2) In the second case, the credential associated to l^+ belongs to a corrupt user, i.e., $j \in \text{CU}$, and we need to consider two subcases. If the adversary wins by satisfying the condition of Step 8 of Figure 3.2, then we have $\mathbf{m}^*(\mathcal{I}) \dashv_{\mathcal{C}} \text{State}_{\mathcal{C}}$. We now formally show this means it is inconsistent with the attributes certified by the organisation and in particular with the ones ($\mathbf{m}$) certified during the l^+ -th issuance query. Regardless of how the l^+ -th query was generated, there exists $\mathcal{I}' \subseteq [m]$ such that $\text{attr}_{l^+} = \mathbf{m}(\mathcal{I}')$. Indeed, if the l^+ -th issuance was performed with a query $\mathcal{O}_{\text{obliss}}(j, \mathbf{m})$ and followed by some corruption query $\mathcal{O}_{\text{CU}}(j)$, then $\mathcal{I}' = [m]$. If on the other hand, it was performed with an $\mathcal{O}_{\text{issue}}(j, \mathbf{m}(\mathcal{I}''))$ query, then $\mathcal{C}$ stored $\text{attr}_{l^+} = \mathbf{m}(\mathcal{I}'')$ and so $\mathcal{I}' = \mathcal{I}''$ fits. Then, by definition of $\dashv_{\mathcal{C}}$, we have that for each $(j', l, \text{cred}_l, \text{attr}_l) \in \text{State}_{\mathcal{C}}$ with $j' \in \text{CU}$, there is at least one index i_l in $\mathcal{I}$ for which $\text{attr}_l[i_l]$ is non-empty and differs from $\mathbf{m}^*(\mathcal{I})[i_l]$. Because $j \in \text{CU}$, this gives the existence of $i_{l^+} \in \mathcal{I}$ such that $\emptyset \neq \text{attr}_{l^+}[i_{l^+}] \neq \mathbf{m}^*(\mathcal{I})[i_{l^+}]$. Said differently, there is some $i_{l^+} \in \mathcal{I}' \cap \mathcal{I}$ such that $\mathbf{m}[i_{l^+}] = \mathbf{m}(\mathcal{I}')[i_{l^+}] = \text{attr}_{l^+}[i_{l^+}] \neq \mathbf{m}^*(\mathcal{I})[i_{l^+}] = \mathbf{m}^*[i_{l^+}]$. Hence, $\mathbf{m}$ and $\mathbf{m}^*$ differ in at least one position.

The second subcase is when the adversary succeeds by performing $\text{ctr}_{\text{cor}} + 1$ inconsistent presentations, each involving a vector of attributes $\mathbf{m}^{(i)}[Z_i]$, for $i \in [\text{ctr}_{\text{cor}} + 1]$. The inconsistency condition of step 10 of Figure 3.2 then directly implies that all the $\mathbf{m}^{(i)}$ are pairwise distinct. As it only owns ctr_{cor} certificates, at least one of these vectors differs from those used during issuance queries and therefore from $\mathbf{m}$. Hence, with probability at least $\frac{1}{\text{ctr}_{\text{cor}} + 1} \geq \frac{1}{Q_{\text{iss}} + 1}$, the corresponding presentation will be selected by the challenger in Game_1 ensuring that $\mathbf{m}^*(\mathcal{I})$ is inconsistent with $\mathbf{m}$.

In both subcases, this formally means that there is at least one index $i_{l^+} \in \mathcal{I}$ such that $\mathbf{m}^*[i_{l^+}] \neq \mathbf{m}[i_{l^+}]$. As a consequence, $\mathbf{x}^*$ features an element $(\mathbf{m}^*[i_{l^+}] - \mathbf{m}[i_{l^+}]) \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{e}_{i_{l^+}}$ and we rely on the same unpredictability argument as (1), but for the i_{l^+} -th column of $(\mathbf{S}_1, \mathbf{S}_2)$. This gives $\text{Adv}_{15}[\mathcal{A}] \leq (Q_{\text{iss}} + 1)(\varepsilon_{\text{M-LWE}}^{(\text{pk})} + \varepsilon_{\text{M-SIS}})$. We note that $\varepsilon_{\text{M-LWE}}^{(\text{pk})} \leq \varepsilon_{\text{M-LWE}}^{(\text{upk})}$ via a simple reduction. Indeed, as the centered binomial $\mathcal{B}_1$ is exactly equal to $U(T_1) - U(T_1)$, an instance $(\mathbf{A}, \mathbf{b})$ of M-LWE $_{n,d,d,q,U(T_1^{2d})}$ can be turned into one of M-LWE $_{n,d,d,q,\mathcal{B}_1^{2d}}$ by computing $\mathbf{b}' = \mathbf{b} - (\mathbf{A}\mathbf{s}' + \mathbf{e}')$ for some $\mathbf{s}', \mathbf{e}' \leftarrow U(T_1^d)$. Adding $(\mathbf{s}', \mathbf{e}')$ to the obtained solution then recovers the secret underlying $\mathbf{b}$.

Hence, in all cases, we have

$$\text{Adv}_{15}[\mathcal{A}] \leq (Q_{\text{iss}} + 1)(\varepsilon_{\text{M-LWE}}^{(\text{upk})} + \varepsilon_{\text{M-SIS}}).$$

Combining all the previous game hops yields the overall result. $\square$

7 Performance and Comparison

We assess the performance of our anonymous credentials by plugging concrete parameters for the sampler and the different zero-knowledge proof systems. The security is then assessed through Theorems 6.1 and 6.2 from the several M-LWE and M-SIS assumptions (which also underly the security of the proof systems). The latter are evaluated using the lattice estimator [APS15]. We select parameters so as to aim for $\lambda = 128$ bits of classical security in the lattice sieving cost model where the hardness bounds of M-LWE and M-SIS are estimated as $2^{-(0.292B+16.4)}$, with B the minimal BKZ blocksize. We also follow the standard practice of not accounting for the loss factors of guessing arguments in the unforgeability proof, as long as the guessing loss is polynomially bounded. For that, we instantiate our scheme with $Q_{\text{iss}} = Q_{\text{show}} = 2^{32}$ which are sufficient for most practical use cases of anonymous credentials⁶. We suggest a complete parameter set in Tables B.1, B.2, B.3, yielding the estimates reported in Table 7.1.

Keys			
$ \text{opk} $	$ \text{osk} $	$ \text{upk} $	$ \text{usk} $
19.03 KB	4.00 KB	2.41 KB	0.25 KB
Transcript	Credential	Show	Security
$ \mathbf{c}, \pi_{\text{iss}}, \text{cred} $	$ \text{cred}_{\text{unb}} $	$ \pi $	$\lambda_{\text{anon}}, \lambda_{\text{uf}}$
29.56 KB	3.52 KB	35.45 KB	128, 126

Table 7.1. Size and security estimates for the anonymous credentials of Section 5. The quantities λ_{anon} and λ_{uf} are the bit-security for the anonymity and unforgeability.

We then compare our scheme to the state-of-the-art constructions of [JS26] and [BLNS23]. In addition to the credential presentation size, i.e., the size of π , we report in Table 7.2 the size of the proof modulus q_π to showcase the improvement of our techniques. Compared to the construction of [JS26], we manage to reduce q_π by roughly 13 bits (and $q_{\pi, \text{iss}}$ by roughly 11 bits for the issuance). Not only does it help improving the proof size, as commitment sizes scale with it, but it also improves the zero-knowledge proof generation and verification runtimes. Indeed, when q_π exceeds 64 bits, operations in $\mathcal{R}_{q_\pi}$ require higher-precision integers (i.e., 128 bits, likely separated into limbs of 64 bits) which slows computations down. Having a modulus q_π fitting in 64 bits then results in smaller runtimes. We however leave the implementation as future work to assess the actual gain.

⁶ Actually, we can likely assume a user will present a credential much less than $2^{32} \approx 4.3$ billion times over the credential’s lifetime. A smaller Q_{show} would require a lower zero-knowledge security, and lead to smaller Prove parameters, and thus smaller presentation proofs π .

Note that this comparison favours the schemes of [JS26] and [BLNS23] as our model is stronger. For example, [BLNS23] would have to be extended with a user secret key to match our model, which would increase its complexity. Despite these discrepancies on the security model, we significantly outperform the scheme of [JS26] by roughly 25%, while still being based on standard lattice assumptions. Our scheme then narrows the gap with the performance showcased in [BLNS23], without resorting to ad-hoc interactive assumptions. We thus believe it provides a strong alternative to this scheme which is particularly well-suited for use-cases like the EUDI wallet where reliance on well-studied assumptions is crucial.

	SK Binding	Blind Iss.	Assumptions	Show Size	$ \pi $	q_π
[BLNS23]	No	Yes	Ad-hoc	122 KB	$\approx 2^{88}$	
			Ad-hoc, Inter.	26 KB	$\approx 2^{50}$	
[JS26]	Yes	No	Standard	46.89 KB	$\approx 2^{65}$	
Ours	Yes	Yes	Standard	35.45 KB	$\approx 2^{52}$	

Table 7.2. Comparison with state-of-the-art anonymous credentials. The “SK Binding” and “Blind Iss.” columns specify if the security model of the scheme supports secret key binding or blind issuance of credentials. The “Assumptions” columns specifies whether or not the underlying lattice assumptions are standard and/or interactive.

References

- AA16. J. Alperin-Sheriff and D. Apon. Dimension-Preserving Reductions from LWE to LWR. *IACR Cryptol. ePrint Arch.*, page 589, 2016.
- AGJ⁺24. S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois, and O. Sanders. Practical Post-Quantum Signatures for Privacy. In *CCS*, 2024.
- APS15. M. R. Albrecht, R. Player, and S. Scott. On the Concrete Hardness of Learning With Errors. *J. Math. Cryptol.*, 2015.
- ASM06. M. H. Au, W. Susilo, and Y. Mu. Constant-Size Dynamic k -TAA. In *SCN*, 2006.
- Ban93. W. Banaszczyk. New Bounds in Some Transference Theorems in the Geometry of Numbers. *Math. Ann.*, 1993.
- Ban95. W. Banaszczyk. Inequalities for Convex Bodies and Polar Reciprocal Lattices in $\mathbb{R}^n$. *Discret. Comput. Geom.*, 13:217–231, 1995.
- BBC⁺24. C. Baum, O. Blazy, J. Camenisch, J. Hoepman, E. Lee, A. Lehmann, A. Lysyanskaya, R. Mayrhofer, H. Montgomery, N. K. Nguyen, B. Peneel, A. Shelat, D. Slamanig, S. Tessaro, S. E. Thomsen, and C. Troncoso. Cryptographers’ Feedback on the EU Digital Identity’s ARF. <https://github.com/user-attachments/files/15904122/cryptographers-feedback.pdf>, 2024.
- BDL⁺18. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More Efficient Commitments from Structured Lattice Assumptions. In *SCN*, 2018.

- 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.
- BJTW26. K. Boudgoust, C. Jeudy, E. Tairi, and W. Wen. Hardness of M-LWE with General Distributions and Applications to Leaky Variants. In *PKC*, 2026.
- BLNS23. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In *CRYPTO*, 2023.
- CL04. J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, 2004.
- DDLL13. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *CRYPTO*, 2013.
- DFJS26. P. Delhom, P.-A. Fouque, C. Jeudy, and O. Sanders. Lattice Group Signatures, Revisited. In *ASIACRYPT*, 2026.
- DLL⁺17. L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Cryptol. ePrint Arch.*, page 633, 2017. Version dated from June 27th 2017.
- dPLS18. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability. In *CCS*, 2018.
- ENI26. ENISA. Agreed Cryptographic Mechanisms. https://certification.enisa.europa.eu/document/download/9993abb8-5f27-47d0-bdb2-f6f284cd5141_en?filename=20260507-acm-draft.pdf, 2026.
- FHS19. G. Fuchsbauer, C. Hanser, and D. Slamanig. Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials. *J. Cryptol.*, 2019.
- GHL22. C. Gentry, S. Halevi, and V. Lyubashevsky. Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties. In *EUROCRYPT*, 2022.
- GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, 2008.
- JRS23. C. Jeudy, A. Roux-Langlois, and O. Sanders. Lattice Signature with Efficient Protocols, Application to Anonymous Credentials. In *CRYPTO*, 2023.
- JS25a. C. Jeudy and O. Sanders. Improved Lattice Blind Signatures from Recycled Entropy. In *CRYPTO*, 2025.
- JS25b. C. Jeudy and O. Sanders. Worst-Case Lattice Sampler with Truncated Gadgets and Applications. In *ASIACRYPT*, 2025.
- JS26. C. Jeudy and O. Sanders. Tag-Friendly Lattice Sampler and Applications. In *ASIACRYPT*, 2026.
- LCL⁺23. Q. Lai, C. Chen, 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.
- Leh26. A. Lehmann. EUDI Wallet & Anonymous Credentials: Status and Open Challenges. https://privcryptworkshop.github.io/slides/key/slides_Lehmann.pdf, 2026.
- LN22. V. Lyubashevsky and N. K. Nguyen. BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications. *ASIACRYPT*, 2022.
- LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. *CRYPTO*, 2022.

- LNPS21. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations. In *ASIACRYPT*, 2021.
- LNS21. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments. In *PKC*, 2021.
- LS15. A. Langlois and D. Stehlé. Worst-case to Average-case Reductions for Module Lattices. *DCC*, 2015.
- 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.
- otEU24. European Parliament & Council of the European Union. Regulation (EU) 2024/1183 of the European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202401183, 2024.
- PS16. D. Pointcheval and O. Sanders. Short Randomizable Signatures. In *CT-RSA*, 2016.
- Sec23. Senior Officials Group Information Systems Security. SOGIS Agreed Cryptographic Mechanisms (version 1.3). https://www.sogis.eu/uk/supporting_doc_en.html, 2023.
- TD24. J. Traoré and A. Dumanois. BBS# and eIDAS 2.0: Making BBS Anonymous Credentials eIDAS 2.0 Compliant. <https://csrc.nist.gov/csrf/media/Presentations/2024/wpec2024-3b3/images-media/wpec2024-3b3-slides-antoine-jacques--BBS-sharp-eIDAS2.pdf>, 2024.
- TZ23. S. Tessaro and C. Zhu. Revisiting BBS signatures. In *EUROCRYPT*, 2023.

A Zero-Knowledge Arguments

In this section, we formally describe our modified zero-knowledge protocols for our anonymous credentials, namely ($\text{Prove}_{\text{iss}}$, $\text{Verify}_{\text{iss}}$) and (Prove , Verify). This section is given for completeness and future reference of the zero-knowledge protocols we use in our construction. For readers familiar with the framework of [LNP22], we note that the main changes incurred by our techniques are located in the conditions on q_π (Sections A.3.6 and A.4.6), and the first round of each protocol (Algorithms A.1 and A.8) in the definition of $\mathbf{g}$, and finally in the number of rows of each ARP component. The other parts of the protocols adapt those formally described in [LNP22], and later works, e.g., [BLNS23, AGJ⁺24, JS25a], to our specific relations.

A.1 Additional Preliminaries

A.1.1 Using Subrings. Following the subring embedding technique formalized in [LNPS21, Sec. 2.8] or [AGJ⁺24, Sec. 2.1] and used in many subsequent constructions, our anonymous credentials system primarily works in the ring $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ and performs the zero-knowledge proofs (in both Issue and Show) in the subring $\mathcal{R}_\pi = \mathbb{Z}[x]/\langle x^{n_\pi} + 1 \rangle$, where $n_\pi | n$ are two powers of two.

We can naturally leverage the tower of power-of-two cyclotomic rings to embed relations over $\mathcal{R}$ into $\mathcal{R}_\pi$. We thus define $k_\pi = n/n_\pi$ as well as the subring embedding $\theta : \mathcal{R} \rightarrow \mathcal{R}_\pi^{k_\pi}$ by

$$\theta \left(\sum_{j=0}^{n-1} a_j x^j \right) = \left[\sum_{j=0}^{n_\pi-1} a_{k_\pi j + i} x^j \right]_{i \in [0, k_\pi-1]} \in \mathcal{R}_\pi^{k_\pi}.$$

Notice that if r belongs to $\mathcal{R}_\pi$, we can lift it to $\mathcal{R}$ as $r(x^{k_\pi})$. In that case, applying θ simplifies to $\theta(r(x^{k_\pi})) = [r|0|\dots|0]^T$. Then, for $\mathbf{b} = [b_i]_{i \in [0, k_\pi-1]} \in \mathcal{R}_\pi^{k_\pi}$, we can define the inverse embedding by $\theta^{-1}(\mathbf{b}) = \sum_{i \in [0, k_\pi-1]} b_i (x^{k_\pi})^i \in \mathcal{R}$, where the multiplication by x^i is performed in $\mathcal{R}$. The embedding transforms the multiplication over $\mathcal{R}$ into a matrix-vector product over $\mathcal{R}_\pi^{k_\pi}$ similarly to the coefficient embedding⁷. More precisely, $\theta(ab) = M_\theta(a)\theta(b) \in \mathcal{R}_\pi^{k_\pi}$ with $M_\theta(a) = [\theta(a)|\theta(ax)|\dots|\theta(ax^{k_\pi-1})]$. The embeddings θ and M_θ extend to vectors and matrices entrywise.

A.1.2 Dilithium Compression. We recall here the decomposition functions used in the framework of [LNP22] to compress the commitments so as to reduce the size of the proofs. For these functions alone, we call $k' = k \bmod^+ q$ the unique element in $[0, q)$ such that $k' = k \bmod q$, in opposition to the centered representative with $\bmod^\pm q$. A complete description of these functions is given in [DLL⁺17, LNP22].

<u>Power2Round_{qπ}(r, D)</u>	<u>Decompose_{qπ}(r, γ)</u>
00 $r \leftarrow r \bmod^+ q_\pi$	10 $r \leftarrow r \bmod^+ q_\pi$
01 $r_L \leftarrow r \bmod^\pm 2^D$	11 $r_L \leftarrow r \bmod^\pm \gamma$
02 return $(r_H, r_L) = ((r - r_L)/2^D, r_L)$	12 if $r - r_L = q_\pi - 1$
<u>UseGHint_{qπ}(y, r, γ)</u>	13 $(r_H, r_L) \leftarrow (0, r_L - 1)$
03 $\alpha \leftarrow (q_\pi - 1)/\gamma$	14 else $r_H \leftarrow (r - r_L)/\gamma$
04 $r_H \leftarrow \text{HighBits}_{q_\pi}(r, \gamma)$	15 return (r_H, r_L)
05 return $(r_H + y) \bmod^+ \alpha$	<u>HighBits_{qπ}(r, γ)</u>
<u>MakeGHint_{qπ}(z, r, γ)</u>	16 $(r_H, r_L) \leftarrow \text{Decompose}_{q_\pi}(r, \gamma)$
06 $\alpha \leftarrow (q_\pi - 1)/\gamma$	17 return r_H
07 $r_H \leftarrow \text{HighBits}_{q_\pi}(r, \gamma)$	
08 $v_H \leftarrow \text{HighBits}_{q_\pi}(r + z, \gamma)$	
09 return $(v_H - r_H) \bmod^\pm \alpha$	

Fig. A.1. Functions for commitment compression.

⁷ The coefficient embedding is actually an edge case of the subring embedding θ when $n_\pi = 1$.

A.2 Parameter Setting

The challenge space $\mathcal{C}$ for the zero-knowledge proofs is defined in accordance with [LNP22]. We let $\rho = \lfloor (2^{2(\lambda+1)/n_\pi} - 1)/2 \rfloor$, and define

$$\mathcal{C} = \left\{ c \in \tau^{-1}([- \rho, \rho]^{n_\pi}) : c^* = c \text{ and } \|c^{64}\|_1^{1/64} \leq \tilde{\eta} \right\},$$

where $\tilde{\eta}$ is chosen so that at least half of the elements c in $\tau^{-1}([- \rho, \rho]^{n_\pi})$ such that $c^* = c$ verify the norm bound. As we choose q_π to be a sufficiently large prime congruent to 5 modulo 8, it ensures that any difference of distinct elements of $\mathcal{C}$ is invertible modulo $q_\pi \mathcal{R}_\pi$. In both proof systems, we let $\ell = \lceil \lambda/2 \log_2 q_\pi \rceil$, and $\gamma, D \in \mathbb{N}$ be parameters for the Dilithium compression such that γ is an even divisor of $q_\pi - 1$ and $D = \max(0, \lfloor 1 + \log_2(\gamma/(\rho n_\pi)) \rfloor)$ so that $2^{D-1} \rho n_\pi \leq \gamma$.

A.3 Issuance Proof

Let us now describe how to prove the desired relation in the issuance protocol [Issue](#). The user proves knowledge the opening of its commitment $\mathbf{c}$, in addition to showing it knows the secret key associated to the given user public key upk . We explain how to transform the relation of Section 5.3 to our protocol ([Prove_{iss}](#), [Verify_{iss}](#)) before describing the latter in details.

A.3.1 Initial Relation. As described in Section 5.3, the issuance proves the following relation. For a more concise notation, for $\mathbf{m} \in \mathcal{R}^m$, $\mathbf{D} = [\mathbf{d}_1 | \dots | \mathbf{d}_m] \in \mathcal{R}^{d \times m}$ and a set $S \subseteq [m]$, we define $\mathbf{m}_S = [\mathbf{m}[i]]_{i \in S}$ and $\mathbf{D}_S = [\mathbf{d}_i]_{i \in S}$. We also let $I = |\mathcal{I}|$ and $I' = |\overline{\mathcal{I}}| = m - I$.

$$\mathbf{r}_1 + \mathbf{A}\mathbf{r}_2 + \mathbf{D}_{\overline{\mathcal{I}}}\mathbf{m}_{\overline{\mathcal{I}}} = \mathbf{c}' := \mathbf{c} - \text{upk} - \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}} \text{ mod } q\mathcal{R} \quad (1a)$$

$$\mathbf{s}_1 + \mathbf{A}\mathbf{s}_2 = \text{upk} \text{ mod } q\mathcal{R} \quad (2a)$$

$$\|\mathbf{r}_i\|_2^2 \leq B_r^2 \text{ for } i \in \{1, 2\} \quad (3a)$$

$$\mathbf{s}_i \in T_1^d \text{ for } i \in \{1, 2\}, \text{ and } \mathbf{m}_{\overline{\mathcal{I}}} \in T_1^{I'} \quad (4a)$$

Although the user proceeds as if it were proving these equations, our optimization of Section 4.2 only allows to extract $(\mathbf{r}_1^*, \mathbf{r}_2^*, \mathbf{m}_{\overline{\mathcal{I}}}^*, \mathbf{s}_1^*, \mathbf{s}_2^*)$ that verify them up to some short additive slacks $(q_\pi \bmod q)\mathbf{h}_{\text{iss},1}$ and $(q_\pi \bmod q)\mathbf{h}_{\text{iss},2}$ for Equations (1a) and (2a) respectively that are also recovered during the extraction. As these slacks only intervene at the end of the soundness proof and are not witnesses whose knowledge must be proven, the actual relation the prover carries the proof with is the one described in Equations (1a), (2a), (3a) and (4a)

A.3.2 Embedding into $\mathcal{R}_\pi$. We start by embedding the relation to be over $\mathcal{R}_\pi$ rather than $\mathcal{R}$. For that, we use the subring embedding θ so that proving the modified equations over $\mathcal{R}_\pi$ is equivalent to the equations above via the inverse embedding. Observing that the coefficient embedding $\tau(\theta(a))$ is equal to

a permuted version of $\tau(a)$, it holds that Equations (3a) and (4a) can directly be expressed in terms of the embeddings. Then, using the multiplication map M_θ , we can transform Equations (1a) and (2a) accordingly. We then obtain

$$\theta(\mathbf{r}_1) + M_\theta(\mathbf{A})\theta(\mathbf{r}_2) + M_\theta(\mathbf{D}_{\overline{\mathcal{T}}})\theta(\mathbf{m}_{\overline{\mathcal{T}}}) = \theta(\mathbf{c}') \bmod q\mathcal{R}_\pi \quad (1b)$$

$$\theta(\mathbf{s}_1) + M_\theta(\mathbf{A})\theta(\mathbf{s}_2) = \theta(\text{upk}) \bmod q\mathcal{R}_\pi \quad (2b)$$

$$\|\theta(\mathbf{r}_i)\|_2^2 \leq B_r^2 \text{ for } i \in \{1, 2\} \quad (3b)$$

$$\theta(\mathbf{s}_i) \in T_1^{dk_\pi} \text{ for } i \in \{1, 2\}, \text{ and } \theta(\mathbf{m}_{\overline{\mathcal{T}}}) \in T_1^{I'k_\pi} \quad (4b)$$

A.3.3 Turning Norm Inequalities into Equalities. To turn the norm inequalities of Equation (3b) into equalities, we extend the witness with a four-square decomposition as described in [BLNS23, AGJ⁺24]. For $i \in \{1, 2\}$, we let $a_{i,0}, \dots, a_{i,3} \in \mathbb{N}$ be such that $B_r^2 - \|\theta(\mathbf{r}_i)\|_2^2 = \sum_{j \in [0,3]} a_{i,j}^2$. We then define $a_i = \sum_{j \in [0,3]} a_{i,j} x^j \in \mathcal{R}_\pi$. These elements are then included in the witness and hidden in the zero-knowledge proof. We then prove the following equation, which indeed implies Equation (3b).

$$\left\| \begin{bmatrix} \theta(\mathbf{r}_i) \\ a_i \end{bmatrix} \right\|_2^2 = B_r^2 \text{ for } i \in \{1, 2\}. \quad (3c)$$

A.3.4 Bimodal Rejection Sampling. We then transform the relation to support bimodal rejection sampling on the witness vector [LN22]. For that, we sample a random sign $\mathbf{b} \in \{-1, 1\}$ beforehand and include it in the witness vector, which then remains hidden throughout the proof. We then change the relation to multiply the entire witness by $\mathbf{b}$ and reexpress all the above equations equivalently in terms the signed witness. We also need to prove that $\mathbf{b}$ is a sign. For the latter it suffices to prove that $\mathbf{b}$, when seen as an element of $\mathcal{R}_\pi$, is a constant polynomial, that is $\langle \tau(\mathbf{b}), \tau(x^i) \rangle = 0$ for all $i \in [1, n_\pi - 1]$, and $\mathbf{b}^2 = 1 \bmod q_\pi \mathcal{R}_\pi$. As q_π is prime, the combination of both statements ensures $\mathbf{b} \in \{-1, 1\}$.

The norms of Equation (3c) are invariant with multiplication by a sign. For the binary constraints of Equation (4b), we instead prove that the signed witness has coefficients in $\{0, \mathbf{b}\}$. Using [LNP22, Lem. 5.2] or rather its generalization to the bimodal case in [JS25a, Lem. A.1], $\tau(\mathbf{b}\mathbf{x}) \in \{0, \mathbf{b}\}^N$ is equivalent to $\langle \tau(\mathbf{b}\mathbf{x}), \tau(\mathbf{b}\mathbf{x} - \mathbf{b}\mathbf{1}) \rangle = 0$, where $\mathbf{1}$ is the vector of $\mathcal{R}_\pi$ whose coefficient embedding is the all-ones vector. With additional restrictions on q_π , the equation can be proven modulo q_π and is thus an evaluation modulo q_π of a quadratic function of the witness. Finally, Equations (1b) and (2b) can also be transformed to be expressed with respect to the signed witness. We then have to prove the following modified equations.

$$\mathbf{b}\theta(\mathbf{r}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{r}_2) + M_\theta(\mathbf{D}_{\overline{\mathcal{T}}})\mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) - \mathbf{b}\theta(\mathbf{c}') = \mathbf{0} \bmod q\mathcal{R}_\pi \quad (1d)$$

$$\mathbf{b}\theta(\mathbf{s}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{s}_2) - \mathbf{b}\theta(\text{upk}) = \mathbf{0} \bmod q\mathcal{R}_\pi \quad (2d)$$

$$\left\| \begin{bmatrix} \mathbf{b}\theta(\mathbf{r}_i) \\ \mathbf{b}a_i \end{bmatrix} \right\|_2^2 = B_r^2 \text{ for } i \in \{1, 2\} \quad (3d)$$

$$\begin{aligned} \tau_0((\mathbf{b}\theta(\mathbf{s}_i))^*(\mathbf{b}\theta(\mathbf{s}_i) - \mathbf{b}\mathbf{1}_{\mathcal{R}_\pi^{dk_\pi}})) &= 0 \bmod q_\pi \mathbb{Z} \text{ for } i \in \{1, 2\}, \\ \text{and } \tau_0((\mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}))^*(\mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) - \mathbf{b}\mathbf{1}_{\mathcal{R}_\pi^{l'k_\pi}})) &= 0 \bmod q_\pi \mathbb{Z} \end{aligned} \quad (4d)$$

$$\mathbf{b}^2 - 1 = 0 \bmod q_\pi \mathcal{R}_\pi, \text{ and } \forall i \in [1, n_\pi - 1], \tau_0(\mathbf{b}x^{n_\pi-i}) = 0 \bmod q_\pi \mathbb{Z} \quad (5d)$$

A.3.5 Lifting. As our modulus q is smooth due to $q = b_H^{k_H} b_L^{k_L}$, it is unsuited to be a divisor of q_π as this would place further arithmetic restrictions on b_H, b_L but it would also drastically hinder soundness. We instead work with a prime q_π and lift Equations (1d) and (2d) to be proven modulo q_π instead. As described in Section 4, it introduces a lifting term $\mathbf{j}$ for each equation. We prove each $\mathbf{j}$ is small, i.e., bounded by some B_{arp} , using the approximate range proof component. As we also need to prove the witness is small for Equations (3d) and (4d), merging the approximate range proofs would entail a constraint in $q_\pi > O(B_{arp}^2)$. The latter is very demanding on the modulus q_π , whereas it comes from trying to prove exact norms for witness that are extremely small. We thus perform two different approximate range proofs, yielding some bound $B_{arp,1}$ on the witness and $B_{arp,2}$ on the lifting terms. Additionally, as described in Section 4.2, we can relax the lifting condition on the modulus q_π to limit the impact of the large term $qB_{arp,2}$. This only incurs additive slacks $\mathbf{h}_{iss,1}, \mathbf{h}_{iss,2}$ in the extracted solution which we account for in the security proof of our anonymous credentials.

More concretely, when writing Equations (1d) and (2d) in $\mathcal{R}_\pi$, we have $\mathbf{j}_1 = q^{-1}(\mathbf{b}\theta(\mathbf{r}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{r}_2) + M_\theta(\mathbf{D}_{\overline{\mathcal{T}}})\mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) - \mathbf{b}\theta(\mathbf{c}'))$ and $\mathbf{j}_2 = q^{-1}(\mathbf{b}\theta(\mathbf{s}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{s}_2) - \mathbf{b}\theta(\text{upk}))$. Using $\|\mathbf{M}\mathbf{x}\|_2 \leq \|\mathbf{M}\|_F \|\mathbf{x}\|_2$, we can deduce the following a priori bound on $\mathbf{j}_1, \mathbf{j}_2$ as

$$\begin{aligned} \|\mathbf{j}_1\|_2 &\leq B_{j,1} := \frac{B_r}{q} + \frac{1}{2}ndB_r + \frac{1}{2}nI'\sqrt{nd} + \frac{1}{2}\sqrt{nd} \\ \|\mathbf{j}_2\|_2 &\leq B_{j,2} := \frac{\sqrt{nd}}{q} + \frac{1}{2}nd\sqrt{nd} + \frac{1}{2}\sqrt{nd} \end{aligned}$$

Based on our optimization of Section 4.3, we need $q_\pi > C_1(128, 128)\sqrt{2nd}B_{arp,2}$ with $B_{arp,2} = \sqrt{\pi/\ln(M_{3,2})} \cdot C_2(128, 128)\sqrt{B_{j,1}^2 + B_{j,2}^2}$. From Table 4.1, we have $C_1(128, 128) \approx 93$ and $C_2(128, 128) \approx 150$. In the end, Equations (1d) and (2d) are implicitly proven with the approximate range proof, and thus transformed into

$$\left\| q^{-1} \begin{bmatrix} \mathbf{b}\theta(\mathbf{r}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{r}_2) + M_\theta(\mathbf{D}_{\overline{\mathcal{T}}})\mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) - \mathbf{b}\theta(\mathbf{c}') \\ \mathbf{b}\theta(\mathbf{s}_1) + M_\theta(\mathbf{A})\mathbf{b}\theta(\mathbf{s}_2) - \mathbf{b}\theta(\text{upk}) \end{bmatrix} \right\|_2 \leq B_{arp,2} \quad (1-2e)$$

A.3.6 Requirements on q_π . Using the methodologies described in Section 4, the final equations require additional constraints on q_π to be verified

in order to imply, in the end, Equations (1a) to (4a). Let us gather all the necessary conditions. First, Equation (3d) with the tighter approximate range proof as detailed in Section 4.3 requires $q_\pi > C_1(128, 128)\sqrt{n_\pi m_1}B_{arp,1}$ and $q_\pi > \max(B_r^2, B_{arp,1}^2 - B_r^2)$, with $B_{arp,1} = \sqrt{\pi/\ln(M_{3,1})} \cdot C_2(128, 128)B_{x1}$ and $B_{x1} = \sqrt{2B_r^2 + 2nd + nI' + 1}$ is a bound on the witness (and m_1 the witness dimension in $\mathcal{R}_\pi$). We also want the binary constraints of Equation (4d) to hold over $\mathbb{Z}$, which imposes $q_\pi > B_{arp,1}^2 + B_{arp,1}\sqrt{nd}$ and $q_\pi > B_{arp,1}^2 + B_{arp,1}\sqrt{nI'}$. Then, the second part of Equation (5d) should also hold over $\mathbb{Z}$, thus requiring $q_\pi > B_{arp,1}$.

Based on the above, the extraction would have proven that $\mathbf{b}$ is indeed a sign, and that $\mathbf{s}_i, \mathbf{r}_i, \mathbf{m}_{\bar{\mathcal{T}}}$ satisfy their respective constraints. Then, for the lifting equation (1-2e), the extractor would recover $(\mathbf{b}\mathbf{r}_i)^*, (\mathbf{b}\mathbf{s}_i)^*, (\mathbf{b}\mathbf{m}_{\bar{\mathcal{T}}})^*, \mathbf{b}^*$ and define the unsigned witnesses $\mathbf{r}_i^*, \mathbf{s}_i^*, \mathbf{m}_{\bar{\mathcal{T}}}^*$. It would then define

$$\begin{aligned} \mathbf{j}_1^* &= q^{-1} (\theta(\mathbf{r}_1^*) + M_\theta(\mathbf{A})\theta(\mathbf{r}_2^*) + M_\theta(\mathbf{D}_{\bar{\mathcal{T}}})\theta(\mathbf{m}_{\bar{\mathcal{T}}}^*) - \theta(\mathbf{c}')) \bmod q_\pi \mathcal{R}_\pi \\ \mathbf{j}_2^* &= q^{-1} (\theta(\mathbf{s}_1^*) + M_\theta(\mathbf{A})\theta(\mathbf{s}_2^*) - \theta(\text{upk})) \bmod q_\pi \mathcal{R}_\pi \end{aligned}$$

and would deduce that

$$\begin{aligned} \mathbf{r}_1^* + \mathbf{A}\mathbf{r}_2^* + \mathbf{D}_{\bar{\mathcal{T}}}\mathbf{m}_{\bar{\mathcal{T}}}^* - \mathbf{c}' - q\theta^{-1}(\mathbf{j}_1^*) &= \mathbf{0} \bmod q_\pi \mathcal{R} \\ \mathbf{s}_1^* + \mathbf{A}\mathbf{s}_2^* - \text{upk} - q\theta^{-1}(\mathbf{j}_2^*) &= \mathbf{0} \bmod q_\pi \mathcal{R} \end{aligned}$$

Rewriting it in $\mathcal{R}_\pi$ computes $\mathbf{h}_{\text{iss},1}, \mathbf{h}_{\text{iss},2}$ such that

$$\begin{aligned} \mathbf{r}_1^* + \mathbf{A}\mathbf{r}_2^* + \mathbf{D}_{\bar{\mathcal{T}}}\mathbf{m}_{\bar{\mathcal{T}}}^* - \mathbf{c}' - q\theta^{-1}(\mathbf{j}_1^*) - q_\pi \mathbf{h}_{\text{iss},1} &= \mathbf{0} \\ \mathbf{s}_1^* + \mathbf{A}\mathbf{s}_2^* - \text{upk} - q\theta^{-1}(\mathbf{j}_2^*) - q_\pi \mathbf{h}_{\text{iss},2} &= \mathbf{0} \end{aligned}$$

Instead of imposing demanding constraints on q_π so as to ensure $\mathbf{h}_{\text{iss},i} = \mathbf{0}$, we use the relaxed lifting of Section 4.2. In particular, we can deduce meaningful bounds on the extracted slack $\mathbf{h}_{\text{iss},i}$. We have $\|\mathbf{h}_{\text{iss},1}\|_2 \leq B_{h,\text{iss},1}$ with

$$B_{h,\text{iss},1} := \frac{1}{q_\pi} \left(B_r + \frac{q}{2}ndB_r + \frac{q}{2}nI'\sqrt{nd} + \frac{q}{2}\sqrt{nd} + qB_{arp,2} \right) = \frac{B_{\mathbf{j},1} + qB_{arp,2}}{q_\pi}$$

and $\|\mathbf{h}_{\text{iss},2}\|_2 \leq B_{h,\text{iss},2}$ with

$$B_{h,\text{iss},2} := \frac{1}{q_\pi} \left(\sqrt{nd} + \frac{q}{2}nd\sqrt{nd} + \frac{q}{2}\sqrt{nd} + qB_{arp,2} \right) = \frac{B_{\mathbf{j},2} + qB_{arp,2}}{q_\pi}$$

Taking the equations modulo q then gives

$$\begin{aligned} \mathbf{r}_1^* + \mathbf{A}\mathbf{r}_2^* + \mathbf{D}_{\bar{\mathcal{T}}}\mathbf{m}_{\bar{\mathcal{T}}}^* - \mathbf{c}' - (q_\pi \bmod q)\mathbf{h}_{\text{iss},1} &= \mathbf{0} \bmod q\mathcal{R} \\ \mathbf{s}_1^* + \mathbf{A}\mathbf{s}_2^* - \text{upk} - (q_\pi \bmod q)\mathbf{h}_{\text{iss},2} &= \mathbf{0} \bmod q\mathcal{R} \end{aligned}$$

as desired by Section 5.3. This imposes no further constraint besides that $(q_\pi \bmod q)B_{h,\text{iss},i}$ should remain small compared to B_{w1} so as not to affect the final M-SIS

bound in our security proof. We thus impose that $q_\pi \bmod q = 1$, but we insist that we could tolerate $q_\pi \bmod q = l$ for some small integer l . Then, we place no further constraint on q_π as our parameters already give $(B_{j,i} + qB_{arp,2})/q_\pi \ll B_{w1}$. In the end, q_π must satisfy the following constraints in addition to being prime and congruent to 5 modulo 8.

$$\begin{aligned} q_\pi &> \max(B_r^2, B_{arp,1}^2 - B_r^2, B_{arp,1}) \\ q_\pi &> B_{arp,1} + B_{arp,1} \sqrt{n \max(d, I')} \quad (\text{Cond}_{\text{iss}}) \\ q_\pi &> C_1(128, 128) \cdot \max(\sqrt{n_\pi m_1} \cdot B_{arp,1}, \sqrt{2nd} \cdot B_{arp,2}) \end{aligned}$$

In our case, the limiting condition of Equation (Cond_{iss}) is the last one, which simplifies to $q_\pi > C_1(128, 128) \sqrt{2nd} B_{arp,2}$.

A.3.7 The Prove_{iss} Protocol. We now give the protocol to prove Equations (1-2e), (3d), (4d) and (5d). To simplify the notations, we define

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{b}\theta(\mathbf{r}_1) \\ \mathbf{b}a_1 \\ \mathbf{b}\theta(\mathbf{r}_2) \\ \mathbf{b}a_2 \\ \mathbf{b}\theta(\mathbf{s}_1) \\ \mathbf{b}\theta(\mathbf{s}_2) \\ \mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) \\ \mathbf{b} \end{bmatrix}, \text{ and } \begin{cases} \mathbf{x}_{1,r,i} = \mathbf{b}\theta(\mathbf{r}_i) & \mathbf{x}'_{1,r,i} = \begin{bmatrix} \mathbf{b}\theta(\mathbf{r}_i) \\ \mathbf{b}a_i \end{bmatrix}, \quad i \in \{1, 2\} \\ \mathbf{x}_{1,s,i} = \mathbf{b}\theta(\mathbf{s}_i), \quad i \in \{1, 2\} \\ \mathbf{x}_{1,m} = \mathbf{b}\theta(\mathbf{m}_{\overline{\mathcal{T}}}) \quad \mathbf{x}_{1,b} = \mathbf{b} \end{cases} \quad (7)$$

The witness dimension is then $m_1 = 2(dk_\pi + 1) + 2dk_\pi + I'k_\pi + 1 = k_\pi(4d + I') + 3$. We also define the following public matrices and vectors.

$$\mathbf{A}_\theta = M_\theta(\mathbf{A}), \quad \mathbf{D}_\theta = M_\theta(\mathbf{D}_{\overline{\mathcal{T}}}), \quad \mathbf{c}'_\theta = \theta(\mathbf{c}'), \quad \text{upk}_\theta = \theta(\text{upk}).$$

The protocol consists of 5 rounds which we detail separately below. They follow the blueprint of [LNP22, LN22], which has been described in details in later works [BLNS23, AGJ⁺24, JS25a], adapted to our relation and implementing our optimizations of Section 4.

Algorithm A.1: Prove_{iss}.Rd₁($\mathbf{x}_1$)

1. $\mathbf{x}_{2,1} \leftarrow \mathcal{B}_1^{m_2 - d_\pi}, \mathbf{x}_{2,2} \leftarrow \mathcal{B}_1^{d_\pi}$
2. $\mathbf{y}_1 \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_1}, \sigma_1}$
3. $\mathbf{y}_{2,1} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_2 - d_\pi}, \sigma_2}, \mathbf{y}_{2,2} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{d_\pi}, \sigma_2}$
4. $\mathbf{y}_{3,1} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{128/n_\pi}, \sigma_{3,1}}, \mathbf{y}_{3,2} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{128/n_\pi}, \sigma_{3,2}}$
5. $\mathbf{g} \leftarrow \text{proj}(\mathbf{B}_g \mathbf{x}_{2,1} \bmod q_\pi \mathcal{R}_\pi) \quad \triangleright \mathbf{g} \in \{a \in \mathcal{R}_{\pi, q_\pi} : \tau_0(a) = \tau_{n_\pi/2}(a) = 0\}^\ell$
6. $\mathbf{t}_A \leftarrow \mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}'_2 \mathbf{x}_{2,1} + \mathbf{x}_{2,2} \bmod q_\pi \mathcal{R}_\pi$
7. $\mathbf{w} \leftarrow \mathbf{A}_1 \mathbf{y}_1 + \mathbf{A}'_2 \mathbf{y}_{2,1} + \mathbf{y}_{2,2} \bmod q_\pi \mathcal{R}_\pi$
8. $\mathbf{t}_{B,y1} \leftarrow \mathbf{B}_{y1} \mathbf{x}_{2,1} + \mathbf{y}_{3,1} \bmod q_\pi \mathcal{R}_\pi$
9. $\mathbf{t}_{B,y2} \leftarrow \mathbf{B}_{y2} \mathbf{x}_{2,1} + \mathbf{y}_{3,2} \bmod q_\pi \mathcal{R}_\pi$
10. $\mathbf{t}_{B,g} \leftarrow \mathbf{B}_g \mathbf{x}_{2,1} + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi$
11. $(\mathbf{w}_H, \mathbf{w}_L) \leftarrow \text{Decompose}_{q_\pi}(\mathbf{w}, \gamma)$
12. $(\mathbf{t}_{A,H}, \mathbf{t}_{A,L}) \leftarrow \text{Power2Round}_{q_\pi}(\mathbf{t}_A, D)$

Output: $(\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}, \mathbf{g}, \mathbf{t}_{A,H}, \mathbf{t}_{A,L}, \mathbf{w}_H, \mathbf{w}_L, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g})$

Algorithm A.2: Prove_{iss}.Rd₂($\mathbf{x}_1, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \mathbf{R}_1, \mathbf{R}_2$)

1. $\tilde{\mathbf{z}}_{3,1} \leftarrow \tau(\mathbf{y}_{3,1}) + \mathbf{R}_1 \cdot \tau(\mathbf{x}_1)$
2. $\mathbf{x}_{1,j} \leftarrow q^{-1} \begin{bmatrix} \mathbf{x}_{1,r,1} + \mathbf{A}_\theta \mathbf{x}_{1,r,2} + \mathbf{D}_\theta \mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{c}'_\theta \\ \mathbf{x}_{1,s,1} + \mathbf{A}_\theta \mathbf{x}_{1,s,2} - \mathbf{x}_{1,b} \text{upk}_\theta \end{bmatrix}$
3. $\tilde{\mathbf{z}}_{3,2} \leftarrow \tau(\mathbf{y}_{3,2}) + \mathbf{R}_2 \cdot \tau(\mathbf{x}_{1,j})$

Output: $(\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{x}_{1,j})$

Algorithm A.3: Prove_{iss}.Rd₃($\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}$)

1. **for** $i \in [2\ell]$ **do**
2. $\text{tmp}_i \leftarrow \sum_{j \in [128]} \gamma_{i,j} (\mathbf{e}_{j,\mathcal{R}_\pi}^* \mathbf{y}_{3,1} + \mathbf{r}_{1,j}^* \mathbf{x}_1 - \tilde{\mathbf{z}}_{3,1}[j]) + \sum_{j \in [128]} \gamma_{i,128+j} (\mathbf{e}_{j,\mathcal{R}_\pi}^* \mathbf{y}_{3,2} + \mathbf{r}_{2,j}^* \mathbf{x}_{1,j} - \tilde{\mathbf{z}}_{3,2}[j]) + \gamma_{i,257} (\mathbf{x}'_{1,r,1} * \mathbf{x}'_{1,r,1} - B_r^2) + \gamma_{i,258} (\mathbf{x}'_{1,r,2} * \mathbf{x}'_{1,r,2} - B_r^2) + \gamma_{i,259} \mathbf{x}_{1,s,1}^* (\mathbf{x}_{1,s,1} - \mathbf{x}_{1,b} \mathbf{1}) + \gamma_{i,260} \mathbf{x}_{1,s,2}^* (\mathbf{x}_{1,s,2} - \mathbf{x}_{1,b} \mathbf{1}) + \gamma_{i,261} \mathbf{x}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + \sum_{j \in [1, n_\pi - 1]} \gamma_{i,261+j} \mathbf{x}_{1,b} x^{n_\pi - j}$
3. **for** $i \in [\ell]$ **do**
4. $f_i \leftarrow \mathbf{g}[i] + 2^{-1}(\text{tmp}_{2i-1} + \text{tmp}_{2i-1}^*) + x^{n_\pi/2} \cdot 2^{-1}(\text{tmp}_{2i} + \text{tmp}_{2i}^*) \bmod q_\pi \mathcal{R}_\pi$

Output: $(f_1, \dots, f_\ell)$

Recall that $\mathbf{e}_{j,\mathcal{R}_\pi} = \tau^{-1}(\mathbf{e}_j)$ where $\mathbf{e}_j$ is the j -th canonical vector with a 1 at position j and 0 elsewhere. In particular, $\mathbf{e}_{j,\mathcal{R}_\pi} \in \mathcal{R}_\pi^{128/n_\pi}$. The vectors $\mathbf{r}_{l,j}$ in the first two sums ($l = 1, 2$) are defined as $\mathbf{r}_{l,j} = \tau^{-1}(\mathbf{R}_l^T \mathbf{e}_j)$, meaning $\tau(\mathbf{r}_{1,j}) \in \{-1, 0, 1\}^{n_\pi m_1}$ corresponds to the j -th row of $\mathbf{R}_1$ and $\tau(\mathbf{r}_{2,j}) \in \{-1, 0, 1\}^{n_\pi \cdot 2dk_\pi}$ corresponds to the j -th row of $\mathbf{R}_2$. Then, by developping the expression of f_i , we can express it as follows:

$$\begin{aligned}
 f_i = & \mathbf{g}[i] + \\
 & + (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) \mathbf{x}'_{1,r,1} * \mathbf{x}'_{1,r,1} \\
 & + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) \mathbf{x}'_{1,r,2} * \mathbf{x}'_{1,r,2} \\
 & + 2^{-1}(\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259}) (\mathbf{x}_{1,s,1}^* (\mathbf{x}_{1,s,1} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,s,1}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,s,1}) \\
 & + 2^{-1}(\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260}) (\mathbf{x}_{1,s,2}^* (\mathbf{x}_{1,s,2} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,s,2}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,s,2}) \\
 & + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261}) (\mathbf{x}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,m}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,m}) \\
 & + 2^{-1}(SR_{1,2i-1}^* + x^{n_\pi/2} SR_{1,2i}^*) \mathbf{x}_1 + 2^{-1}(SR_{1,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{x}_1^* \\
 & + 2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{x}_{1,j} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{x}_{1,j}^* \\
 & + 2^{-1}(SE_{1,2i-1}^* + x^{n_\pi/2} SE_{1,2i}^*) \mathbf{y}_{3,1} + 2^{-1}(SE_{1,2i-1} + x^{n_\pi/2} SE_{1,2i}) \mathbf{y}_{3,1}^* \\
 & + 2^{-1}(SE_{2,2i-1}^* + x^{n_\pi/2} SE_{2,2i}^*) \mathbf{y}_{3,2} + 2^{-1}(SE_{2,2i-1} + x^{n_\pi/2} SE_{2,2i}) \mathbf{y}_{3,2}^* \\
 & + c_i \mathbf{x}_{1,b} + c_i^* \mathbf{x}_{1,b}^* \\
 & - d_i
 \end{aligned}$$

where for $i^* \in [2\ell]$,

$$\begin{aligned}
 SE_{1,i^*} &= \sum_{j \in [128]} \gamma_{i^*,j} \mathbf{e}_j \in \mathcal{R}_{\pi,q_\pi}^{128/n_\pi}, \quad SR_{1,i^*} = \sum_{j \in [128]} \gamma_{i^*,j} \mathbf{r}_{1,j} \in \mathcal{R}_{\pi,q_\pi}^{m_1}, \\
 SE_{2,i^*} &= \sum_{j \in [128]} \gamma_{i^*,128+j} \mathbf{e}_j \in \mathcal{R}_{\pi,q_\pi}^{128/n_\pi}, \quad SR_{2,i^*} = \sum_{j \in [128]} \gamma_{i^*,128+j} \mathbf{r}_{2,j} \in \mathcal{R}_{\pi,q_\pi}^{2dk_\pi},
 \end{aligned} \tag{8}$$

and the polynomials c_i, c'_i and d_i for $i \in [\ell]$ are defined by

$$c_i = 2^{-1} \left(-\gamma_{2i,261+n_\pi/2} + \sum_{j=1}^{n_\pi/2-1} (\gamma_{2i-1,261+n_\pi-j} - \gamma_{2i,261+n_\pi/2-j})x^j \right. \\ \left. + \gamma_{2i-1,261+n_\pi/2}x^{n_\pi/2} + \sum_{j=n_\pi/2+1}^{n_\pi-1} (\gamma_{2i-1,261+n_\pi-j} + \gamma_{2i,261+3n_\pi/2-j})x^j \right) \quad (9)$$

$$c'_i = 2^{-1} \left(\gamma_{2i,261+n_\pi/2} + \sum_{j=1}^{n_\pi/2-1} (\gamma_{2i,261+n_\pi/2+j} - \gamma_{2i-1,261+j})x^j - \gamma_{2i-1,261+n_\pi/2}x^{n_\pi/2} \right. \\ \left. - \sum_{j=n_\pi/2+1}^{n_\pi-1} (\gamma_{2i-1,261+j} + \gamma_{2i,261+j-n_\pi/2})x^j \right) \quad (10)$$

$$d_i = \left(\sum_{j \in [128]} \gamma_{2i-1,j} \tilde{\mathbf{z}}_{3,1}[j] + \sum_{j \in [128]} \gamma_{2i-1,128+j} \tilde{\mathbf{z}}_{3,2}[j] + (\gamma_{2i-1,257} + \gamma_{2i-1,258})B_r^2 \right) \\ + x^{n_\pi/2} \left(\sum_{j \in [128]} \gamma_{2i,j} \tilde{\mathbf{z}}_{3,1}[j] + \sum_{j \in [128]} \gamma_{2i,128+j} \tilde{\mathbf{z}}_{3,2}[j] + (\gamma_{2i,257} + \gamma_{2i,258})B_r^2 \right) \quad (11)$$

Algorithm A.4: Prove_{iss}.Rd₄($\mathbf{x}_1, \mathbf{x}_{2,1}, \mathbf{y}_1, \mathbf{y}_{2,1}; \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}, (\mu_i)_i$)

1. $\widehat{\mathbf{m}}_{y1} \leftarrow -\mathbf{B}_{y1}\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
2. $\widehat{\mathbf{m}}_{y2} \leftarrow -\mathbf{B}_{y2}\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
3. $\widehat{\mathbf{m}}_g \leftarrow -\mathbf{B}_g\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
4. Compute $SE_{1,i}, SE_{2,i}, SR_{1,i}, SR_{2,i}$ for $i \in [2\ell]$ and c_i, c'_i for $i \in [\ell]$ as in Equations (8), (9), (10).
5. $\mathbf{y}_{1,j} \leftarrow (q^{-1} \bmod q_\pi) \left[\mathbf{y}_{1,r,1} + \mathbf{A}_\theta \mathbf{y}_{1,r,2} + \mathbf{D}_\theta \mathbf{y}_{1,m} - \mathbf{y}_{1,b} \mathbf{c}'_\theta \right] \bmod q_\pi \mathcal{R}_\pi$
6. $e_0 \leftarrow (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) \mathbf{y}'_{1,r,1} \mathbf{y}'_{1,r,1} + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) \mathbf{y}'_{1,r,2} \mathbf{y}'_{1,r,2} + 2^{-1}(\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259}) (\mathbf{y}_{1,s,1}^* (\mathbf{y}_{1,s,1} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,s,1}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,s,1}) + 2^{-1}(\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260}) (\mathbf{y}_{1,s,2}^* (\mathbf{y}_{1,s,2} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,s,2}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,s,2}) + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261}) (\mathbf{y}_{1,m}^* (\mathbf{y}_{1,m} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,m}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,m}) + \mu_{\ell+1} \mathbf{y}_{1,b}^2$
7. $e_1 \leftarrow \sum_{i \in [\ell]} \mu_i (\widehat{\mathbf{m}}_g[i] + 2^{-1} (SR_{1,2i-1}^* + x^{n_\pi/2} SR_{1,2i}^*) \mathbf{y}_1 + 2^{-1} (SR_{1,2i-1} + x^{n_\pi/2} SR_{1,2i}) \mathbf{y}_1^* + 2^{-1} (SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{y}_{1,j} + 2^{-1} (SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{y}_{1,j}^* + 2^{-1} (SE_{1,2i-1}^* + x^{n_\pi/2} SE_{1,2i}^*) \widehat{\mathbf{m}}_{y1} + 2^{-1} (SE_{1,2i-1} + x^{n_\pi/2} SE_{1,2i}) \widehat{\mathbf{m}}_{y1}^* + 2^{-1} (SE_{2,2i-1}^* + x^{n_\pi/2} SE_{2,2i}^*) \widehat{\mathbf{m}}_{y2} + 2^{-1} (SE_{2,2i-1} + x^{n_\pi/2} SE_{2,2i}) \widehat{\mathbf{m}}_{y2}^* + c_i \mathbf{y}_{1,b} + c'_i \mathbf{y}_{1,b}^* + (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) (\mathbf{y}'_{1,r,1} \mathbf{x}'_{1,r,1} + \mathbf{x}'_{1,r,1} \mathbf{y}'_{1,r,1}) + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) (\mathbf{y}'_{1,r,2} \mathbf{x}'_{1,r,2} + \mathbf{x}'_{1,r,2} \mathbf{y}'_{1,r,2}) + 2^{-1} (\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259}) (\mathbf{y}_{1,s,1}^* (\mathbf{x}_{1,s,1} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,s,1}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,s,1} + \mathbf{x}_{1,s,1}^* (\mathbf{y}_{1,s,1} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,s,1}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,s,1}) + 2^{-1} (\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260}) (\mathbf{y}_{1,s,2}^* (\mathbf{x}_{1,s,2} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,s,2}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,s,2} + \mathbf{x}_{1,s,2}^* (\mathbf{y}_{1,s,2} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,s,2}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,s,2}) + 2^{-1} (\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261}) (\mathbf{y}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,m}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,m} + \mathbf{x}_{1,m}^* (\mathbf{y}_{1,m} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,m}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,m})) + 2\mu_{\ell+1} \mathbf{x}_{1,b} \mathbf{y}_{1,b}$
8. $t_0 \leftarrow \mathbf{b}^T \mathbf{y}_{2,1} + e_0 \bmod q_\pi \mathcal{R}_\pi$
9. $t_1 \leftarrow \mathbf{b}^T \mathbf{x}_{2,1} + e_1 \bmod q_\pi \mathcal{R}_\pi$

Output: (t_0, t_1)

Algorithm A.5: Prove.Rd₅ $\left(\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \right. \\ \left. \mathbf{t}_{A,L}, \mathbf{w}_L, \mathbf{w}_H; \mathbf{R}_1, \mathbf{R}_2, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, c \right)$

1. $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + c\mathbf{x}_1$
2. $\mathbf{z}_{2,1} \leftarrow \mathbf{y}_{2,1} + c\mathbf{x}_{2,1}$ and $\mathbf{z}'_{2,2} \leftarrow \mathbf{y}_{2,2} + c\mathbf{x}_{2,2}$
3. $\mathbf{z}'_2 \leftarrow \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}'_{2,2} \end{bmatrix}$, and $\mathbf{x}_2 \leftarrow \begin{bmatrix} \mathbf{x}_{2,1} \\ \mathbf{x}_{2,2} \end{bmatrix}$
4. $u_1, u_2 \leftarrow U([0, 1])$
5. $\text{rej}_1 \leftarrow \left\| u_1 > \frac{\exp\left(\pi \frac{\|\mathbf{c}\mathbf{x}_1\|_2^2}{\sigma_1^2} + \pi \frac{\|\mathbf{R}_1 \tau(\mathbf{x}_1)\|_2^2}{\sigma_{3,1}^2} + \pi \frac{\|\mathbf{R}_2 \tau(\mathbf{x}_{1,j})\|_2^2}{\sigma_{3,2}^2}\right)}{M_1 M_{3,1} M_{3,2} \cosh\left(2\pi \frac{\langle \tau(\mathbf{z}_1), \tau(c\mathbf{x}_1) \rangle}{\sigma_1^2} + 2\pi \frac{\langle \tilde{\mathbf{z}}_{3,1}, \mathbf{R}_1 \tau(\mathbf{x}_1) \rangle}{\sigma_{3,1}^2} + 2\pi \frac{\langle \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_2 \tau(\mathbf{x}_{1,j}) \rangle}{\sigma_{3,2}^2}\right)} \right\|$
6. $\text{rej}_2 \leftarrow \left\| u_2 > \frac{\exp(\pi \|\mathbf{c}\mathbf{x}_2\|_2^2 / \sigma_2^2)}{M_2 \cosh(2\pi \langle \tau(\mathbf{z}'_2), \tau(c\mathbf{x}_2) \rangle / \sigma_2^2)} \right\|$
7. $\mathbf{z}_{2,2} \leftarrow \mathbf{z}'_{2,2} - c\mathbf{t}_{A,L} - \mathbf{w}_L$
8. $\text{rej}_{\text{comp}} \leftarrow \left\| \|\mathbf{z}_{2,1}\|_2^2 + \|\mathbf{z}_{2,2}\|_2^2 > B_{\pi,2} \right\|$
9. $\mathbf{h} \leftarrow \text{MakeGHint}_{q\pi}(\mathbf{z}_{2,2}, \gamma \mathbf{w}_H - \mathbf{z}_{2,2}, \gamma)$

Output: $(\text{rej}_1, \text{rej}_2, \text{rej}_{\text{comp}}, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$

Algorithm A.6: Prove_{iss}(statement_{iss}, $\mathbf{x}_1$)

1. $(\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}, \mathbf{g}, \mathbf{t}_{A,H}, \mathbf{t}_{A,L}, \mathbf{w}_H, \mathbf{w}_L, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}) \leftarrow \text{Prove}_{\text{iss}}.\text{Rd}_1(\mathbf{x}_1)$
2. $\text{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \mathbf{w}_H)$
3. $(\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, \mathbf{R}_{2,0}, \mathbf{R}_{2,1}) \leftarrow \mathcal{H}(1, \text{statement}_{\text{iss}}, \text{msg}_1) \in (\{0, 1\}^{128 \times 2n\pi m_1})^2 \times (\{0, 1\}^{128 \times 2n\pi d})^2$
4. $\mathbf{R}_1 \leftarrow \mathbf{R}_{1,0} - \mathbf{R}_{1,1}, \mathbf{R}_2 \leftarrow \mathbf{R}_{2,0} - \mathbf{R}_{2,1}$
5. $(\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{x}_{1,j}) \leftarrow \text{Prove}_{\text{iss}}.\text{Rd}_2(\mathbf{x}_1, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \mathbf{R}_1, \mathbf{R}_2)$
6. $\text{msg}_2 = (\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2})$
7. $(\gamma_{i,j})_{\substack{i \in [2\ell] \\ j \in [260+n\pi]}} \leftarrow \mathcal{H}(2, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2) \in \mathbb{Z}_{q\pi}^{2\ell \times (260+n\pi)}$
8. $(f_1, \dots, f_\ell) \leftarrow \text{Prove}_{\text{iss}}.\text{Rd}_3(\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j})$
9. $\text{msg}_3 = (f_1, \dots, f_\ell)$
10. $(\mu_i)_{i \in [\ell+1]} \leftarrow \mathcal{H}(3, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2, \text{msg}_3) \in \mathcal{R}_{\pi, q\pi}^{\ell+1}$
11. $(t_0, t_1) \leftarrow \text{Prove}_{\text{iss}}.\text{Rd}_4(\mathbf{x}_1, \mathbf{x}_{2,1}, \mathbf{y}_1, \mathbf{y}_{2,1}; \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}, (\mu_i)_i)$
12. $\text{msg}_4 = (t_0, t_1)$
13. $c \leftarrow \mathcal{H}(4, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2, \text{msg}_3, \text{msg}_4) \in \mathbb{C}$
14. $(\text{rej}_1, \text{rej}_2, \text{rej}_{\text{comp}}, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h}) \leftarrow \text{Prove}_{\text{iss}}.\text{Rd}_5(\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{t}_{A,L}, \mathbf{w}_L, \mathbf{w}_H; \mathbf{R}_1, \mathbf{R}_2, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, c)$
15. if $[\text{rej}_1 = 1] \vee [\text{rej}_2 = 1] \vee [\text{rej}_{\text{comp}} = 1]$ goto 1.

Output: $\pi_{\text{iss}} = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, f_1, \dots, f_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$

A.3.8 Verification. The verification algorithm $\text{Verify}_{\text{iss}}$ recomputes certain elements (the commitments $\mathbf{w}_H$ and t_0) that are necessary to verify the proof. The verification bounds $B_{\pi,1}, B_{z3,1}, B_{z3,2}$ are determined by the Gaussian tail bound of Lemma 2.2, and $B_{\pi,2}$, which is also used in $\text{Prove}_{\text{iss}}.\text{Rd}_5$, uses both the Gaussian tail bound and the compression norm bounds. In the end, we have

$$\begin{cases} B_{\pi,1} = c_{n\pi m_1} \sigma_1 \sqrt{n_\pi m_1} \\ B_{\pi,2} = c_{n\pi m_2} \sigma_2 \sqrt{n_\pi m_2} + (\tilde{\eta} 2^{D-1} + \frac{\gamma}{2}) \sqrt{n_\pi d_\pi} \\ B_{z3,1} = c_{128 \sigma_{3,1}} \sqrt{128} \\ B_{z3,2} = c_{128 \sigma_{3,2}} \sqrt{128} \end{cases}$$

Algorithm A.7: Verify_{iss}(statement_{iss}, π_{iss})

1. Parse $\pi_{\text{iss}} = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, f_1, \dots, f_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$
2. $\text{tmp} \leftarrow \mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}'_2 \mathbf{z}_{2,1} - c 2^D \mathbf{t}_{A,H} \bmod q_\pi \mathcal{R}_\pi$
3. $\mathbf{w}_H \leftarrow \text{UseGHint}_{q\pi}(\mathbf{h}, \text{tmp}, \gamma)$
4. $\mathbf{z}_{2,2} \leftarrow \gamma \mathbf{w}_H - \text{tmp}$

5. $\mathbf{z}_2 \leftarrow \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}_{2,2} \end{bmatrix}$
6. $\text{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \mathbf{w}_H)$
7. $\text{msg}_2 = (\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2})$
8. $\text{msg}_3 = (f_1, \dots, f_\ell)$
9. $(\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, \mathbf{R}_{2,0}, \mathbf{R}_{2,1}) \leftarrow \mathcal{H}(1, \text{statement}_{\text{iss}}, \text{msg}_1)$
10. $(\mathbf{R}_1, \mathbf{R}_2) \leftarrow (\mathbf{R}_{1,0} - \mathbf{R}_{1,1}, \mathbf{R}_{2,0} - \mathbf{R}_{2,1})$
11. $(\gamma_{i,j})_{\substack{i \in [2\ell] \\ j \in [260+n_\pi]}} \leftarrow \mathcal{H}(2, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2)$
12. $(\mu_i)_{i \in [\ell+1]} \leftarrow \mathcal{H}(3, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2, \text{msg}_3)$
13. $\widehat{\mathbf{m}}_{zy1} \leftarrow \mathbf{c} \mathbf{t}_{B,y1} - \mathbf{B}_{y1} \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
14. $\widehat{\mathbf{m}}_{zy2} \leftarrow \mathbf{c} \mathbf{t}_{B,y1} - \mathbf{B}_{y2} \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
15. $\widehat{\mathbf{m}}_{zg} \leftarrow \mathbf{c} \mathbf{t}_{B,g} - \mathbf{B}_g \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
16. Compute $SE_{1,i}, SE_{2,i}, SR_{1,i}, SR_{2,i}$ for $i \in [2\ell]$ and c_i, c'_i, d_i for $i \in [\ell]$ as in Equations (8), (9), (10), (11)
17. $\mathbf{z}_{1,j} \leftarrow (q^{-1} \bmod q_\pi) \left[\mathbf{z}_{1,r,1} + \mathbf{A}_\theta \mathbf{z}_{1,r,2} + \mathbf{D}_\theta \mathbf{z}_{1,m} - \mathbf{z}_{1,b} \mathbf{c}'_\theta \right] \bmod q_\pi \mathcal{R}_\pi$
18. $t_0 \leftarrow (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) \mathbf{z}'_{1,r,1} + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) \mathbf{z}'_{1,r,2} + 2^{-1}(\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259})(\mathbf{z}'_{1,s,1}(\mathbf{z}_{1,s,1} - \mathbf{z}_{1,b} \mathbf{1}) + (\mathbf{z}'_{1,s,1} - \mathbf{z}_{1,b} \mathbf{1}^*) \mathbf{z}_{1,s,1}) + 2^{-1}(\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260})(\mathbf{z}'_{1,s,2}(\mathbf{z}_{1,s,2} - \mathbf{z}_{1,b} \mathbf{1}) + (\mathbf{z}'_{1,s,2} - \mathbf{z}_{1,b} \mathbf{1}^*) \mathbf{z}_{1,s,2}) + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261})(\mathbf{z}'_{1,m}(\mathbf{z}_{1,m} - \mathbf{z}_{1,b} \mathbf{1}) + (\mathbf{z}'_{1,m} - \mathbf{z}_{1,b} \mathbf{1}^*) \mathbf{z}_{1,m})) + \mu_{\ell+1} \mathbf{z}_{1,b}^2 + c \cdot (\sum_{i \in [\ell]} \mu_i (\widehat{\mathbf{m}}_{zg}[i] + 2^{-1}(SR_{1,2i-1}^* + x^{n_\pi/2} SR_{1,2i}^*) \mathbf{z}_1 + 2^{-1}(SR_{1,2i-1} + x^{n_\pi/2} SR_{1,2i}) \mathbf{z}_1^* + 2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{z}_{1,j} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{z}_{1,j}^* + 2^{-1}(SE_{1,2i-1}^* + x^{n_\pi/2} SE_{1,2i}^*) \widehat{\mathbf{m}}_{zy1} + 2^{-1}(SE_{1,2i-1} + x^{n_\pi/2} SE_{1,2i}) \widehat{\mathbf{m}}_{zy1}^* + 2^{-1}(SE_{2,2i-1}^* + x^{n_\pi/2} SE_{2,2i}^*) \widehat{\mathbf{m}}_{zy2} + 2^{-1}(SE_{2,2i-1} + x^{n_\pi/2} SE_{2,2i}) \widehat{\mathbf{m}}_{zy2}^* + c_i \mathbf{z}_{1,b} + c'_i \mathbf{z}_{1,b}^*)) - c^2 \cdot (\sum_{i \in [\ell]} \mu_i (d_i + f_i) + \mu_{\ell+1}) - c \cdot t_1 + \mathbf{b}^T \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
19. $\text{msg}_4 \leftarrow (t_0, t_1)$
20. $c' \leftarrow \mathcal{H}(4, \text{statement}_{\text{iss}}, \text{msg}_1, \text{msg}_2, \text{msg}_3, \text{msg}_4)$
21. $b \leftarrow \llbracket c = c' \rrbracket \wedge \llbracket \|\mathbf{z}_1\|_2 \leq B_{\pi,1} \rrbracket \wedge \llbracket \|\mathbf{z}_2\|_2 \leq B_{\pi,2} \rrbracket \wedge \llbracket \|\tilde{\mathbf{z}}_{3,1}\|_2 \leq B_{z3,1} \rrbracket \wedge \llbracket \|\tilde{\mathbf{z}}_{3,2}\|_2 \leq B_{z3,2} \rrbracket \wedge \llbracket \|\mathbf{h}\|_\infty \leq \frac{q_\pi - 1}{2\gamma} \rrbracket \wedge \llbracket \forall i \in [\ell], \tau_0(f_i) = \tau_{n_\pi/2}(f_i) = 0 \rrbracket$

Output: b (1 if valid, 0 otherwise)

A.3.9 Security. We now give the formal security result for the proof system ($\text{Prove}_{\text{iss}}, \text{Verify}_{\text{iss}}$) based on the framework (and security) from [LNP22, LN22]. The latter are slightly adapted to embark our optimization of Section 4.

Lemma A.1 (Security of Issuance Proof). *Let $\rho = \lfloor (2^{2(\lambda+1)/n_\pi} - 1)/2 \rfloor$, and $\tilde{\eta} \in \mathbb{N}^\times$ be defining the challenge space $\mathcal{C}$ as*

$$\mathcal{C} = \left\{ c \in \mathcal{R}_\pi : c^* = c \wedge \|c\|_\infty \leq \rho \wedge \|c^{64}\|_1^{1/64} \leq \tilde{\eta} \right\}.$$

Let $M_1, M_2, M_{3,1}, M_{3,2}$ be in $(1, \infty)$ be rejection rates, and let $\alpha_x = \sqrt{\pi / \ln M_x}$ for $x \in \{1, 2, (3, 1), (3, 2)\}$ be the associated rejection sampling slacks. We define the witness $\mathbf{x}_1$ as in Equation (7), which is of dimension $m_1 = k_\pi(4d + I') + 3$ in $\mathcal{R}_\pi$. We let $B_{x1} = \sqrt{2B_r^2 + 2nd + nI' + 1}$ be a bound on $\|\mathbf{x}_1\|_2$. We also define the lifting bounds

$$B_{j,1} = \frac{B_r}{q} + \frac{\sqrt{nd}}{2}(\sqrt{nd}B_r + nI' + 1), \quad \text{and} \quad B_{j,2} = \frac{\sqrt{nd}}{q} + \frac{\sqrt{nd}}{2}(nd + 1),$$

necessary when lifting the mod- q modulo q_π . We then define the Gaussian widths of the masked openings by $\sigma_1 = \alpha_1 \tilde{\eta} B_{x1}$, $\sigma_2 = \alpha_2 \tilde{\eta} \sqrt{n_\pi m_2}$ for $m_2 \in \mathbb{N}^\times$, $\sigma_{3,1} =$

$\alpha_{3,1}\sqrt{\beta(128,128)}B_{x1}$ and $\sigma_{3,2} = \alpha_{3,2}\sqrt{\beta(128,128)}\sqrt{B_{j,1}^2 + B_{j,2}^2}$, where $\beta(\cdot, \cdot)$ is the function defined in Lemma 4.1. We also define approximate range proof bounds $B_{arp,1} = \alpha_{3,1}C_2(128,128)B_{x1}$, $B_{arp,2} = \alpha_{3,2}C_2(128,128)\sqrt{B_{j,1}^2 + B_{j,2}^2}$. Then, let d_π, ℓ, q_π be in $\mathbb{N}^\times$ such that q_π is prime with $q_\pi = 5 \bmod 8$ and satisfies the conditions of Equation (Cond_{iss}), and $\ell = \lceil \lambda/2 \log_2 q_\pi \rceil$. For the Dilithium compression of commitments, we also let γ be an even divisor of $q_\pi - 1$, and $D = \max(0, \lfloor 1 + \log_2(\gamma/(pn_\pi)) \rfloor)$ so that $2^{D-1}pn_\pi \leq \gamma$. Then, (Prove_{iss}, Verify_{iss}) is knowledge sound with an expected-polynomial-time extractor with soundness error

$$\varepsilon_{snd,iss} = \frac{2}{|C|} + q_\pi^{-n_\pi/2} + q_\pi^{-2\ell} + 2^{-\lambda} + \varepsilon_{M-SIS}^{(snd,iss)}$$

and zero-knowledge with loss $\varepsilon_{zk,iss} = \varepsilon_{M-LWE}^{(zk,iss)} + \text{negl}(\lambda)$. The loss $\varepsilon_{M-SIS}^{(snd,iss)}$ is the hardness bound of $M-SIS_{n_\pi, d_\pi, m_1+m_2, q_\pi, \beta_{iss}}$ where β_{iss} is defined as

$$\beta_{iss} = 4\tilde{\eta}\sqrt{4c_{n_\pi m_1}^2 \sigma_1^2 n_\pi m_1 + \left(2c_{n_\pi m_2} \sigma_2 \sqrt{n_\pi m_2} + (2^D \tilde{\eta} + \gamma)\sqrt{n_\pi d_\pi}\right)^2},$$

while $\varepsilon_{M-LWE}^{(zk,iss)}$ is the hardness bound of $M-LWE_{n_\pi, m_2-d_\pi-256/n_\pi-\ell-1, m_2, q_\pi, B_1, B_1}$.

Proof. As the completeness and soundness proofs rigorously follow from that of [LNP22] by simply substituting our new modular Johnson-Lindenstrauss result of Lemma 4.1, we only show the zero-knowledge property to show the definition change of $\mathbf{g}$ does not impact it. For that, we show by a sequence of game hops that valid transcripts are simulatable without resorting to the witness $\mathbf{x}_1$ in a way that is indistinguishable from legitimate proofs. As only the commitment simulation game differs from previous works, e.g., [LNP22, LN22, BLNS23, AGJ⁺24], we may refer to these works for the other arguments. We denote by $\text{Adv}_i[\mathcal{A}]$ the advantage of an adversary to distinguish a legitimate proof from a simulated one as described in Game_{*i*}. In Game₀, the simulator generates the proof legitimately using the witness, which means $\text{Adv}_0 = 0$ by design.

Game₁. In this game, the simulator simulates the random oracle queries by using lazy sampling, i.e., when queried, it returns the corresponding output if the query was already made, and samples a fresh one otherwise. This affects steps 3, 7, 10 and 13 of Prove_{iss}. In the random oracle model, we have

$$\text{Adv}_0[\mathcal{A}] = \text{Adv}_1[\mathcal{A}].$$

Game₂. Here, the simulator samples the final challenge c before step 1 of Prove_{iss}, and reprograms the ROM call of step 13 to output c . For a polynomial number of ROM queries, rejection sampling repetitions and simulator queries, the same argument as in, e.g., [BLNS23] gives

$$|\text{Adv}_1[\mathcal{A}] - \text{Adv}_2[\mathcal{A}]| \leq \text{negl}(n_\pi d_\pi) = \text{negl}(\lambda).$$

Game₃. Then, using the challenge c sampled at the outset, the simulator constructs the openings $\mathbf{z}_1, \mathbf{z}_{2,1}$ directly in Prove_{iss}.Rd₁. After step 3, the simulator

defines $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{x}_1$, $\mathbf{z}_{2,1} = \mathbf{y}_{2,1} + c\mathbf{x}_{2,1}$ and $\mathbf{z}'_{2,2} = \mathbf{y}_{2,2} + c\mathbf{x}_{2,2}$. It then defines $\mathbf{w} = \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}'_2\mathbf{z}_{2,1} + \mathbf{z}'_{2,2} - c\mathbf{t}_A$ in step 7.

Further, in [Prove_{iss}.Rd₄](#), the simulator adjusts some of the computations to only use $\mathbf{x}_{2,1}, \mathbf{z}_{2,1}$ and not $\mathbf{y}_{2,1}$. First, in step 1 to 3 of [Prove_{iss}.Rd₄](#), we define $\hat{\mathbf{m}}_{yi} = c\mathbf{B}_{yi}\mathbf{x}_{2,1} - \mathbf{B}_{yi}\mathbf{z}_{2,1} \bmod q_\pi\mathcal{R}_\pi$ and $\hat{\mathbf{m}}_g = c\mathbf{B}_g\mathbf{x}_{2,1} - \mathbf{B}_g\mathbf{z}_{2,1} \bmod q_\pi\mathcal{R}_\pi$. Then, later instead of performing steps 6 and 8 of [Prove_{iss}.Rd₄](#), the simulator uses $\mathbf{z}_1, \mathbf{z}_{2,1}, c$ and defines t_0 exactly as in step 18 of [Verify_{iss}](#).

By construction and definition of the verification, both games are identical. We then obtain

$$\text{Adv}_2[\mathcal{A}] = \text{Adv}_3[\mathcal{A}].$$

Game₄. Notice that $\mathbf{y}_{2,1}, \mathbf{y}_{2,2}$ are now used only to define $\mathbf{z}_{2,1}, \mathbf{z}'_{2,2}$, but everything else is simulated using $\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{z}_{2,1}, \mathbf{z}'_{2,2}$ directly. The simulator then simulates $\mathbf{z}_{2,1}$ and $\mathbf{z}'_{2,2}$ as follows. To properly set $\mathbf{z}'_2$ (and thus $\mathbf{z}_{2,1}, \mathbf{z}'_{2,2}$ as per step 3 of [Prove_{iss}.Rd₅](#)) and rej_2 in step 8 of [Prove_{iss}.Rd₅](#), we follow the bimodal rejection technique of [LN22, Fig.5, Lem. 3.1]. It samples $\tilde{\mathbf{z}}'_2 \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_2}, \sigma_2}$.

It then sets $\mathbf{z}'_2 = \text{sign}(\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle) \cdot \tilde{\mathbf{z}}'_2$ with probability p and $\mathbf{z}'_2 = -\text{sign}(\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle) \cdot \tilde{\mathbf{z}}'_2$ with probability $1 - p$ for

$$p = \frac{\exp\left(\frac{4\pi\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle}{\sigma_2^2}\right)}{\exp\left(\frac{4\pi\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle}{\sigma_2^2}\right) + 1}$$

It then sets $\text{rej}_2 = 0$ with probability $1/M_2$, and $\text{rej}_2 = 1$ otherwise. By [LN22, Lem. 3.1] (adapted to our definition of Gaussian which differ by a factor $\sqrt{2\pi}$ in the parameter), it holds that $\mathbf{z}'_2$ is identically distributed to that of the previous game. As such, so are $\mathbf{z}_{2,1}$ and $\mathbf{z}'_{2,2}$ as $\mathbf{z}'_2 = [\mathbf{z}'_{2,1} | \mathbf{z}'_{2,2}]^T$. Hence

$$\text{Adv}_3[\mathcal{A}] = \text{Adv}_4[\mathcal{A}].$$

As noted in [LN22], what is important here is that all these steps can be simulated with $\tilde{\mathbf{z}}'_2$, which is sampled at the outset, and $\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle$.

Game₅. We now simulate the ABDLOP commitment using M-LWE (or an extended version due to the bimodal optimization [LN22]). This is where we ever so slightly update the simulation to our new definition of $\mathbf{g}$. In this game, the simulator samples everything as usual (including $\mathbf{x}_2 = [\mathbf{x}_{2,1}^T | \mathbf{x}_{2,2}^T]^T$) from steps 1, 2 and 4 of [Prove_{iss}.Rd₁](#) (step 3 is removed in the previous game) but also $\mathbf{u} \leftarrow U(\mathcal{R}_{\pi, q_\pi}^{d_\pi + 2 \cdot 128/n_\pi + \ell + 1})$. It parses $\mathbf{u}$ as $[\mathbf{u}_{t_A}^T | \mathbf{u}_{y1}^T | \mathbf{u}_{y2}^T | \mathbf{u}_g^T | u_{t_1}]^T$ and defines

$$\begin{aligned} \mathbf{g} &= -\text{proj}(\mathbf{u}_g) \\ \mathbf{t}_A &= \mathbf{u}_{t_A} + \mathbf{A}_1\mathbf{x}_1 \bmod q_\pi\mathcal{R}_\pi \\ \mathbf{t}_{B,y1} &= \mathbf{u}_{y1} + \mathbf{y}_{3,1} \bmod q_\pi\mathcal{R}_\pi \\ \mathbf{t}_{B,y2} &= \mathbf{u}_{y2} + \mathbf{y}_{3,2} \bmod q_\pi\mathcal{R}_\pi \\ \mathbf{t}_{B,g} &= \mathbf{u}_g + \mathbf{g} \bmod q_\pi\mathcal{R}_\pi \\ t_1 &= u_{t_1} + e_1 \bmod q_\pi\mathcal{R}_\pi \end{aligned}$$

It also uses $\mathbf{u}_{y1}, \mathbf{u}_{y2}, \mathbf{u}_g$ to define $\hat{\mathbf{m}}_{y1}, \hat{\mathbf{m}}_{y2}, \hat{\mathbf{m}}_g$ as per the modification in **Game₃**:

$$\begin{aligned}\hat{\mathbf{m}}_{y1} &= c\mathbf{u}_{y1} - \mathbf{B}_{y1}\mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi \\ \hat{\mathbf{m}}_{y2} &= c\mathbf{u}_{y2} - \mathbf{B}_{y2}\mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi \\ \hat{\mathbf{m}}_g &= c\mathbf{u}_g - \mathbf{B}_g\mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi\end{aligned}$$

It still uses $\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle$ to set $\mathbf{z}'_2$ as in the previous game. We now argue that distinguishing **Game₄** from **Game₅** comes down to distinguishing an extended (knapsack) M-LWE instance from random. Indeed, given the instance

$$\left(\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{A}'_2 & \mathbf{I}_{d_\pi} \\ \mathbf{B}_{y1} & \mathbf{0} \\ \mathbf{B}_{y2} & \mathbf{0} \\ \mathbf{B}_g & \mathbf{0} \\ \mathbf{b}^T & \mathbf{0} \end{bmatrix}, \mathbf{u}, \tilde{\mathbf{z}}'_2, \langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle \right),$$

the reduction uses these elements to run the simulator of **Game₅**. Then, if $\mathbf{u} = \bar{\mathbf{B}}\mathbf{x}_2 \bmod q_\pi \mathcal{R}_\pi$, this perfectly simulates **Game₄** (in particular as $\mathbf{u}_g = \mathbf{B}_g\mathbf{x}_{2,1} \bmod q_\pi \mathcal{R}_\pi$). On the other hand, if $\mathbf{u}$ is uniform, then it perfectly simulates **Game₅**. As a result, if $\varepsilon_{\text{M-LWE}}^{(zk, \text{iss})}$ denotes the hardness bound of this extended (knapsack) M-LWE problem, we have

$$|\text{Adv}_4[\mathcal{A}] - \text{Adv}_5[\mathcal{A}]| \leq \varepsilon_{\text{M-LWE}}^{(zk, \text{iss})}.$$

As noted in [LN22], reductions from M-LWE to such extended versions exist in the literature [AA16, LNS21, BJRW23, BJTW26]. The hint being a single short integer, [BJTW26] typically show it would reduce the hardness bound by at worse a factor $|\langle \tau(\tilde{\mathbf{z}}'_2), \tau(c\mathbf{x}_2) \rangle|$, simply by guessing this value. As it is usual to omit small polynomial guessing loss factors, we simply estimate $\varepsilon_{\text{M-LWE}}^{(zk, \text{iss})}$ through the hardness bound of M-LWE $_{n_\pi, m_2 - d_\pi - 256/n_\pi - \ell - 1, m_2, q_\pi, \mathcal{B}_1, \mathcal{B}_1}$.

Game₆. We now simulate specifically the commitment t_1 to ensure the simulator does need to compute $\hat{\mathbf{m}}_{y1}, \hat{\mathbf{m}}_{y2}, \hat{\mathbf{m}}_g$ (which depend on $\mathbf{u}_{y1}, \mathbf{u}_{y2}, \mathbf{u}_g$). The simulator then selects $t_1 \leftarrow U(\mathcal{R}_{\pi, q_\pi})$. Because u_{t_1} is uniform and perfectly independent of e_1 in **Game₅**, it holds that t_1 was already uniform, then matching **Game₆**. Hence

$$\text{Adv}_5[\mathcal{A}] = \text{Adv}_6[\mathcal{A}].$$

Note that at this stage, the simulator no longer needs to compute e_1 , and therefore does not need to use $\mathbf{u}_{y1}, \mathbf{u}_{y2}, \mathbf{u}_g$ in **Prove_{iss}.Rd₄**.

Game₇. We further simulate the commitment, and in particular decouple the sampling of $\mathbf{t}_{B,g}$ and $\mathbf{g}$. More precisely, the simulator now samples $\mathbf{t}_A \leftarrow U(\mathcal{R}_{\pi, q_\pi}^{d_\pi})$, and $\mathbf{t}_{B,y1}, \mathbf{t}_{B,y2} \leftarrow U(\mathcal{R}_{\pi, q_\pi}^{128/n_\pi})$. It also samples

$$\begin{aligned}\mathbf{g} &\leftarrow U(\{g \in \mathcal{R}_{\pi, q_\pi} : \tau_0(g) = \tau_{n_\pi/2}(g) = 0\}^\ell) \\ \text{and } \mathbf{t}_{B,g} &\leftarrow U(\{r + r'x^{n_\pi/2} : (r, r') \in \mathbb{Z}_{q_\pi}^2\}^\ell).\end{aligned}$$

We now argue both games are identical. First, in **Game**₆, it holds that $\mathbf{u}_{t_A}, \mathbf{u}_{y_1}, \mathbf{u}_{y_2}$ are uniform and independent (among themselves and) of $\mathbf{A}_1 \mathbf{x}_1, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}$ respectively, and are used nowhere else. Hence, $\mathbf{t}_A, \mathbf{t}_{B,y_1}, \mathbf{t}_{B,y_2}$ in **Game**₆ are uniform in their respective space, thus matching **Game**₇. For $\mathbf{g}$ and $\mathbf{t}_{B,g}$, we have in **Game**₆ that

$$\mathbf{g} = -\text{proj}(\mathbf{u}_g) = - \left[\sum_{j \neq 0, n_\pi/2} \tau_j(\mathbf{u}_g[i]) x^j \right]_{i \in [\ell]}$$

$$\text{and } \mathbf{t}_{B,g} = \mathbf{u}_g - \text{proj}(\mathbf{u}_g) = \left[\tau_0(\mathbf{u}_g[i]) + \tau_{n_\pi/2}(\mathbf{u}_g[i]) x^{n_\pi/2} \right]_{i \in [\ell]}$$

As all coefficients of each entry of $\mathbf{u}_g$ are uniform and independent, it indeed holds that $\mathbf{g}$ and $\mathbf{t}_{B,g}$ in **Game**₆ are independent and uniform in their respective space. As a result, **Game**₆ and **Game**₇ are identical, yielding

$$\text{Adv}_6[\mathcal{A}] = \text{Adv}_7[\mathcal{A}].$$

The rest of the proof is carried as in previous works, leveraging that $\mathbf{g}$ is now uniform in the polynomials with 0 coefficients in degree 0 and $n_\pi/2$.

Game₈. We now simulate the f_i as the $\mathbf{g}[i]$ now act as a perfect mask for them. The simulator then samples each $f_i \leftarrow U(\{g \in \mathcal{R}_{\pi, q_\pi} : \tau_0(g) = \tau_{n_\pi/2}(g) = 0\})$. Because $\mathbf{g}[i]$ in this space and independent of the rest, and because f_i must satisfy $\tau_0(f_i) = \tau_{n_\pi/2}(f_i) = 0$ for valid proofs, it holds that both games are identical. As a result, we get

$$\text{Adv}_7[\mathcal{A}] = \text{Adv}_8[\mathcal{A}].$$

Game₉. At this stage, $\mathbf{x}_1$ is used nowhere else than in constructing $\mathbf{z}_1$ and $\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}$. We thus simulate all of them at once using the bimodal rejection sampling. Indeed, because $\mathbf{x}_1 = \mathbf{b}\mathbf{x}'_1$, $\tau(\mathbf{x}_1) = \mathbf{b}\tau(\mathbf{x}'_1)$ and $\tau(\mathbf{x}_{1,j}) = \mathbf{b}\tau(\mathbf{x}'_{1,j})$ with $\mathbf{b}$ the bimodal sign (used nowhere else as perfectly hidden by the simulated commitments) and with $\mathbf{x}'_1, \tau(\mathbf{x}'_1), \tau(\mathbf{x}'_{1,j})$ independent of $\mathbf{b}$, we can apply bimodal rejection sampling, e.g., [LN22, Lem. 2.8 (2)]. In particular, the simulator samples $\mathbf{z}_1 \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_1}, \sigma_1}$, $\tilde{\mathbf{z}}_{3,1} \leftarrow \mathcal{D}_{\mathbb{Z}^{128}, \sigma_{3,1}}$ and $\tilde{\mathbf{z}}_{3,2} \leftarrow \mathcal{D}_{\mathbb{Z}^{128}, \sigma_{3,2}}$. By [LN22, Lem. 2.8 (2)] which is attributed to [DDLL13], both games are identical, thus giving

$$\text{Adv}_8[\mathcal{A}] = \text{Adv}_9[\mathcal{A}].$$

The simulator in **Game**₉ does not use the witness $\mathbf{x}_1$ whatsoever. Combining all the advantage equations then yields

$$\text{Adv}_9[\mathcal{A}] \leq \varepsilon_{\text{M-LWE}}^{(zk, \text{iss})} + \text{negl}(\lambda),$$

as claimed. $\square$

A.3.10 Proof Size. Using the rANS encoding for discrete Gaussian vectors, we can compute the size of the zero-knowledge proof π_{iss} produced by `Proveiss` using the following formula. Compared to [LNP22], we now have two ARP vectors $\tilde{\mathbf{z}}_{3,1}$ and $\tilde{\mathbf{z}}_{3,2}$ instead of one but each with a reduced dimension from 256 to 128 thanks to our optimized modular Johnson-Lindenstrauss result. This also transpires in the commitments $\mathbf{t}_{B,y1}$ and $\mathbf{t}_{B,y2}$. Finally, the commitment $\mathbf{t}_{B,g}$ can now be compressed to 2ℓ integers modulo q_π as the coefficients of each entry of $\mathbf{t}_{B,g}$ are zeros except for the coefficients of degree 0 and $n_\pi/2$. All the parameters are set according to Lemma A.1. This leads to

$$\begin{aligned} |\pi_{\text{iss}}| &= n_\pi d_\pi (\lceil \log_2 q_\pi \rceil - D + 2.25) + (256 + 2\ell + \ell n_\pi + n_\pi) \lceil \log_2 q_\pi \rceil \\ &\quad + n_\pi \lceil \log_2(2\rho + 1) \rceil + n_\pi m_1 \left(\frac{1}{2} + \log_2 \sigma_1 \right) + n_\pi (m_2 - d_\pi) \left(\frac{1}{2} + \log_2 \sigma_2 \right) \\ &\quad + 128 \left(\frac{1}{2} + \log_2 \sigma_{3,1} \right) + 128 \left(\frac{1}{2} + \log_2 \sigma_{3,2} \right). \end{aligned}$$

A.4 Credential Presentation Proof

We proceed with detailing the credential presentation proof needed in `Show`. The user generates such a proof at each presentation of its credential. We present the modified relation we effectively prove in our protocol and how it allows us to conclude on the original one. We then give the full description of (`Prove`, `Verify`).

A.4.1 Initial Relation. Just like for the issuance proof, the relation the user actually proves slightly differs from the equations presented in Section 5.3. This is because our relaxed lifting technique of Section 4.2 only allows to extract the witness *and* a small additive slack that appear in the proven equations. In this case, it extracts $(\mathbf{w}_1^*, \mathbf{w}_2^*, \mathbf{v}_3^*, t^*, \mathbf{m}^*)$ such that $\mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + [q_L t^* \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3^* = \mathbf{u} + \mathbf{D}\mathbf{m}^* + (q_\pi \bmod q)\mathbf{h} \bmod q\mathcal{R}$, for some short $\mathbf{h}$ that is also recovered during the extraction. Again, as $\mathbf{h}$ only intervenes at the end of the soundness proof and is not a witness whose knowledge must be proven, the actual relation the prover carries the proof with is

$$\mathbf{w}_1 + \mathbf{A}\mathbf{w}_2 + [q_L r(x^{k_\pi}) \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3 - \mathbf{D}_{\overline{\mathcal{I}}}\mathbf{m}_{\overline{\mathcal{I}}} = \mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}} \bmod q\mathcal{R} \quad (6a)$$

$$\|\mathbf{w}_i\|_2^2 \leq B_{wi}^2 \text{ for } i \in \{1, 2\}, \text{ and } \|\mathbf{v}_3\|_2^2 \leq B_3^2 \quad (7a)$$

$$\|r\|_2^2 = \omega \text{ and } \tau(r) \in \{0, 1\}^{n_\pi}, \text{ and } \mathbf{m}_{\overline{\mathcal{I}}} \in T_1^{I'} \quad (8a)$$

Due to the structure of our tag space $\mathcal{T}_w$, the tag t is of the form $r(x^{k_\pi})$ for some $r \in \mathcal{R}_\pi$ with binary coefficient and Hamming weight ω . As a result, following the strategy of [LNP22, Sec. 6.3], we can simply prove knowledge of r (an element of degree n_π) rather than the sparse element t (of degree $n = k_\pi n_\pi$). As we later embed the relation in $\mathcal{R}_\pi$, this allows for proving tag membership more efficiently. We use the same notations $\mathbf{D}_S, \mathbf{m}_S$ and $I' = m - |\mathcal{I}|$ as in the issuance proof of Section A.3, but we recall that the set of disclosed attribute indices $\mathcal{I}$ may differ from that of the issuance, and also between different showings. For clarity in the rest of the description, we define $\mathbf{u}' = \mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}} \bmod q\mathcal{R}$.

A.4.2 Embedding into $\mathcal{R}_\pi$. As in Section A.3.2, we can use the subring embedding θ from $\mathcal{R}$ to $\mathcal{R}_\pi^{k_\pi}$ to work directly over $\mathcal{R}_\pi$. Every term is as straightforward to embed as in the issuance using the multiplication map M_θ , except for the quadratic term multiplying $t = r(x^{k_\pi})$ and $\mathbf{v}_3$. We rewrite this term as

$$[q_L r(x^{k_\pi}) \mathbf{G}_H | \mathbf{0}] \mathbf{v}_3 = \sum_{i=0}^{k_H-1} q_L b_H^i r(x^{k_\pi}) \mathbf{v}_{3,i},$$

where each $\mathbf{v}_{3,i} \in \mathcal{R}^d$ and $\mathbf{v}_3 = [\mathbf{v}_{3,0}^T | \dots | \mathbf{v}_{3,k_H-1}^T | v_{3,a}]^T$, with $v_{3,a}$ matching the column $\mathbf{a}$ (which does not intervene in this quadratic term). Applying θ to this quadratic term leads to expressing $\theta(r(x^{k_\pi}) \mathbf{v}_{3,i})$. Notice that $\theta(r(x^{k_\pi})) = [r | \mathbf{0}]^T \in \mathcal{R}_\pi^{k_\pi}$. Hence, when applying θ to each of the d entries of $r(x^{k_\pi}) \mathbf{v}_{3,i}$, we have

$$\theta(r(x^{k_\pi}) \cdot \mathbf{v}_{3,i}[j]) = M_\theta(\mathbf{v}_{3,i}[j]) \begin{bmatrix} r \\ \mathbf{0} \end{bmatrix} = r \theta(\mathbf{v}_{3,i}[j]).$$

When applied on the whole vector, it then yields

$$\theta([q_L r(x^{k_\pi}) \mathbf{G}_H | \mathbf{0}] \mathbf{v}_3) = r \sum_{i=0}^{k_H-1} q_L b_H^i \theta(\mathbf{v}_{3,i}) = r M_\theta([q_L \mathbf{G}_H | \mathbf{0}]) \theta(\mathbf{v}_3).$$

Note that $M_\theta([q_L \mathbf{G}_H | \mathbf{0}])$ simplifies to $[q_L (\mathbf{G}_H \otimes \mathbf{I}_{k_\pi}) | \mathbf{0}_{dk_\pi \times k_\pi}]$. Overall, Equations (6a), (7a), (8a) become

$$\begin{aligned} & \theta(\mathbf{w}_1) + M_\theta(\mathbf{A}) \theta(\mathbf{w}_2) + r M_\theta([q_L \mathbf{G}_H | \mathbf{0}]) \theta(\mathbf{v}_3) + M_\theta([- \mathbf{B} | \mathbf{a}]) \theta(\mathbf{v}_3) \\ & - M_\theta(\mathbf{D}_{\bar{\mathcal{T}}}) \theta(\mathbf{m}_{\bar{\mathcal{T}}}) = \theta(\mathbf{u}') \bmod q \mathcal{R}_\pi \end{aligned} \quad (6b)$$

$$\|\theta(\mathbf{w}_i)\|_2^2 \leq B_{w_i}^2 \text{ for } i \in \{1, 2\}, \text{ and } \|\theta(\mathbf{v}_3)\|_2^2 \leq B_3^2 \quad (7b)$$

$$\|r\|_2^2 = \omega, \text{ and } r \in T_1, \text{ and } \theta(\mathbf{m}_{\bar{\mathcal{T}}}) \in T_1^{I' k_\pi} \quad (8b)$$

A.4.3 Turning Norm Inequalities into Equalities. As in Section A.3.3, we use four-square decompositions to turn Equation (7b) to norm equalities. We thus define $a_1, a_2, a_3 \in \mathcal{R}_\pi$ to be degree 3 polynomials containing respectively the four-square decomposition of $B_{w1}^2 - \|\theta(\mathbf{w}_1)\|_2^2$, $B_{w2}^2 - \|\theta(\mathbf{w}_2)\|_2^2$ and $B_3^2 - \|\theta(\mathbf{v}_3)\|_2^2$. We then prove the following equation, which indeed implies Equation (7b).

$$\left\| \begin{bmatrix} \theta(\mathbf{w}_i) \\ a_i \end{bmatrix} \right\|_2^2 = B_{w_i}^2 \text{ for } i \in \{1, 2\}, \text{ and } \left\| \begin{bmatrix} \theta(\mathbf{v}_3) \\ a_3 \end{bmatrix} \right\|_2^2 \leq B_3^2 \quad (7c)$$

A.4.4 Bimodal Rejection Sampling. As in Section A.3.4, we change the relation to be able to perform bimodal rejection sampling, which requires to encompass the sign $\mathbf{b} \in \{-1, 1\}$ beforehand. Concretely, we multiply the entire witness by $\mathbf{b}$ and append $\mathbf{b}$ to the witness. We proceed in the same way as before for proving norm and binary constraints. For equation (6b), we proceed slightly differently than in the issuance proof. In the latter, the relation was purely linear,

meaning that multiplying it entirely by $\mathbf{b}$ gave an equivalent relation that is linear in the signed witness. In the show proof, the quadratic term would become cubic when multiplied by $\mathbf{b}$. We instead use the fact that $\mathbf{b}^2 = 1$ to obtain the following equivalent relation.

$$\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_1) + M_\theta(\mathbf{A})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_2) + \mathbf{br}M_\theta([q_L \mathbf{G}_H | \mathbf{0}])\mathbf{b}\theta(\mathbf{v}_3) + M_\theta([- \mathbf{B} | \mathbf{a}])\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{v}_3) - M_\theta(\mathbf{D}_{\bar{\mathcal{T}}})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{m}_{\bar{\mathcal{T}}}) = \theta(\mathbf{u}') \bmod q\mathcal{R}_\pi \quad (6d)$$

$$\left\| \begin{bmatrix} \mathbf{b}\theta(\mathbf{w}_i) \\ \mathbf{b}a_i \end{bmatrix} \right\|_2^2 = B_{w_i}^2 \text{ for } i \in \{1, 2\}, \text{ and } \left\| \begin{bmatrix} \mathbf{b}\theta(\mathbf{v}_3) \\ \mathbf{b}a_3 \end{bmatrix} \right\|_2^2 \leq B_3^2 \quad (7d)$$

$$\|\mathbf{br}\|_2^2 = \omega, \text{ and } \tau_0((\mathbf{br})^*(\mathbf{br} - \mathbf{b}\mathbf{1}_{\mathcal{R}_\pi^{k_\pi}})) = 0 \bmod q\mathbb{Z}, \quad (8d)$$

$$\text{and } \tau_0((\mathbf{b}\theta(\mathbf{m}_{\bar{\mathcal{T}}}))^*(\mathbf{b}\theta(\mathbf{m}_{\bar{\mathcal{T}}}) - \mathbf{b}\mathbf{1}_{\mathcal{R}_\pi^{I'k_\pi}})) = 0 \bmod q_\pi\mathbb{Z}$$

$$\mathbf{b}^2 - 1 = 0 \bmod q_\pi\mathcal{R}_\pi, \text{ and } \forall i \in [1, n_\pi - 1], \tau_0(\mathbf{b}x^{n_\pi-i}) = 0 \bmod q_\pi\mathbb{Z} \quad (9d)$$

A.4.5 Lifting. For the same reasons as in the issuance proof, our modulus q requires a specific lifting to be able to prove the equation modulo q_π instead. We proceed exactly as in Section A.3.5, writing the equation without modular reduction with a lifting term $\mathbf{j}$ as

$$q\mathbf{j} = \mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_1) + M_\theta(\mathbf{A})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_2) + \mathbf{br}M_\theta([q_L \mathbf{G}_H | \mathbf{0}])\mathbf{b}\theta(\mathbf{v}_3) + M_\theta([- \mathbf{B} | \mathbf{a}])\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{v}_3) - M_\theta(\mathbf{D}_{\bar{\mathcal{T}}})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{m}_{\bar{\mathcal{T}}}) - \theta(\mathbf{u}')$$

Noting the bimodal sign and the embedding θ or θ^{-1} do not affect the norm, we can deduce a bound on $\mathbf{j}$ through the simpler equation $q\theta^{-1}(\mathbf{j}) = \mathbf{w}_1 + \mathbf{A}\mathbf{w}_2 + \sum_{i=0}^{k_H-1} q_L b_H^i r(x^{k_\pi})\mathbf{v}_{3,i} + [- \mathbf{B} | \mathbf{a}]\mathbf{v}_3 - \mathbf{D}_{\bar{\mathcal{T}}}\mathbf{m}_{\bar{\mathcal{T}}} - \mathbf{u}'$. In particular, we get $\|\mathbf{j}\|_2 \leq B_j$ with

$$B_j := \frac{B_{w1}}{q} + \frac{nd}{2}B_{w2} + \omega\sqrt{\frac{1 - 1/q_H^2}{b_H^2 - 1}}B_3 + \frac{n\sqrt{d(dk_H + 1)}}{2}B_3 + \frac{nI'\sqrt{nd}}{2} + \frac{\sqrt{nd}}{2}$$

As in the issuance proof, we do not commit to $\mathbf{j}$ in the witness but rather prove it is short using the approximate range proof. As such, we need $q_\pi > C_1(128, 128)\sqrt{nd}B_{arp,2}$ with $B_{arp,2} = \sqrt{\pi/\ln(M_{3,2})} \cdot C_2(128, 128)B_j$. Hence, we then transform Equation (6d) into

$$\|q^{-1}(\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_1) + M_\theta(\mathbf{A})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{w}_2) + \mathbf{br}M_\theta([q_L \mathbf{G}_H | \mathbf{0}])\mathbf{b}\theta(\mathbf{v}_3) + M_\theta([- \mathbf{B} | \mathbf{a}])\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{v}_3) - M_\theta(\mathbf{D}_{\bar{\mathcal{T}}})\mathbf{b} \cdot \mathbf{b}\theta(\mathbf{m}_{\bar{\mathcal{T}}}) - \theta(\mathbf{u}'))\|_2 \leq B_{arp,2}. \quad (6e)$$

A.4.6 Requirements on q_π . To deduce the modified equations indeed prove the desired relation, we again need to impose some lower bounds on q_π . First, Equation (7d) and the first part of Equation (8d) with the tighter approximate range proof as detailed in Section 4.3 requires $q_\pi > C_1(128, 128)\sqrt{n_\pi m_1}B_{arp,1}$ and $q_\pi > \max(B_{w1}^2, B_{w2}^2, B_3^2, \omega)$ and $q_\pi > B_{arp,1}^2 - \min(B_{w1}^2, B_{w2}^2, B_3^2, \omega)$, with $B_{arp,1} = \sqrt{\pi/\ln(M_{3,1})} \cdot C_2(128, 128)B_{x1}$ and B_{x1} is a bound on the witness

defined by $B_{x1} = \sqrt{B_{w1}^2 + B_{w2}^2 + B_3^2 + \omega + nI' + 1}$ (and m_1 the witness dimension in $\mathcal{R}_\pi$). For the binary constraint of Equation (8d), the one on the attribute vector follows as in the issuance proof, requiring $q_\pi > B_{arp,1}^2 + B_{arp,1}\sqrt{nI'}$. That on the tag is however simplified due to the exact norm proven on r , which then gives $q_\pi > \omega + \sqrt{\omega n}$. Then, the second part of Equation (9d) should also hold over $\mathbb{Z}$, thus requiring $q_\pi > B_{arp,1}$.

There remains only Equation (6e). Following the same extraction strategy as in the issuance proof of Section A.3.6, one can extract the sign $\mathbf{b}$ and the signed witness, and define the unsigned witness $(\mathbf{w}_1^*, \mathbf{w}_2^*, \mathbf{v}_3^*, r^*, \mathbf{m}_{\mathcal{I}}^*)$. It would then define $\theta^{-1}(\mathbf{j}^*) = q^{-1}(\mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + \sum_{i=0}^{k_H-1} q_L b_H^i r^*(x^{k_\pi}) \mathbf{v}_{3,i}^* + [-\mathbf{B}|\mathbf{a}]\mathbf{v}_3^* - \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}^* - \mathbf{u}') \bmod q_\pi \mathcal{R}$. Rewriting it in $\mathcal{R}_\pi$ computes $\mathbf{h}^*$ such that

$$\mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + [q_L t^* \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3^* - \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}^* - \mathbf{u}' - q\theta^{-1}(\mathbf{j}^*) - q_\pi \mathbf{h}^* = \mathbf{0},$$

with $t^* = r^*(x^{k_\pi})$. It then holds $\|\mathbf{h}^*\|_2 \leq B_h$ with

$$\begin{aligned} B_h &:= \frac{1}{q_\pi} \left(\frac{B_{w1}}{q} + \frac{nd}{2} B_{w2} + \omega \sqrt{\frac{1 - 1/q_H^2}{b_H^2 - 1}} B_3 + \frac{n\sqrt{d(dk_H + 1)}}{2} B_3 \right. \\ &\quad \left. + \frac{nI'\sqrt{nd}}{2} + \frac{\sqrt{nd}}{2} + qB_{arp} \right) \\ &= \frac{1}{q_\pi} (B_{\mathbf{j}} + qB_{arp}) \end{aligned}$$

Taking the equation modulo q then gives

$$\mathbf{w}_1^* + \mathbf{A}\mathbf{w}_2^* + [q_L t^* \mathbf{G}_H - \mathbf{B}|\mathbf{a}]\mathbf{v}_3^* - \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}^* - (q_\pi \bmod q) \mathbf{h}^* = \mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}} \bmod q\mathcal{R},$$

as desired in Section 5.3. This imposes no further constraint besides that $(q_\pi \bmod q)(B_{\mathbf{j}} + qB_{arp})/q_\pi$ should remain sufficiently small compared to B_{w1} so as not to affect the final M-SIS bound in our security proof. We thus impose that $q_\pi \bmod q = 1$, but we insist that we could tolerate $q_\pi \bmod q = l$ for some small integer l . Then, we place no further constraint on q_π (beyond the condition $q_\pi > C_1(128, 128)\sqrt{nd}B_{arp,2}$ to apply Lemma 4.1) as our parameters already give $(B_{\mathbf{j}} + qB_{arp})/q_\pi \ll B_{w1}$. In the end, q_π must satisfy the following constraints in addition to being prime and congruent to 5 modulo 8, and congruent to 1 modulo q .

$$\begin{aligned} q_\pi &> \max(B_{w1}^2, B_{w2}^2, B_3^2, \omega) \\ q_\pi &> B_{arp,1}^2 - \min(B_{w1}^2, B_{w2}^2, B_3^2, \omega) \\ q_\pi &> B_{arp,1}^2 + B_{arp,1}\sqrt{nI'} & (\text{Cond}_{\text{show}}) \\ q_\pi &> \omega + \sqrt{\omega n} \\ q_\pi &> C_1(128, 128) \cdot \max(\sqrt{n_\pi m_1} \cdot B_{arp,1}, \sqrt{nd} \cdot B_{arp,2}) \end{aligned}$$

In our case, the limiting condition of Equation (Cond_{show}) is $q_\pi > B_{arp,1}^2 + B_{arp,1}\sqrt{nI'}$.

A.4.7 The Prove Protocol. We now give the protocol to prove Equations (6e), (7d), (8d), and (9d). We again simplify the notations with

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{b}\theta(\mathbf{w}_1) \\ \mathbf{b}a_1 \\ \mathbf{b}\theta(\mathbf{w}_2) \\ \mathbf{b}a_2 \\ \mathbf{b}\theta(\mathbf{v}_3) \\ a_3 \\ \mathbf{b}\theta(t) \\ \mathbf{b}\theta(\mathbf{m}_{\bar{T}}) \\ \mathbf{b} \end{bmatrix}, \text{ and } \begin{cases} \mathbf{x}_{1,w,i} = \mathbf{b}\theta(\mathbf{w}_i), \mathbf{x}'_{1,w,i} = \begin{bmatrix} \mathbf{b}\theta(\mathbf{w}_i) \\ \mathbf{b}a_i \end{bmatrix}, i \in \{1, 2\} \\ \mathbf{x}_{1,v} = \mathbf{b}\theta(\mathbf{v}_3), \mathbf{x}'_{1,v} = \begin{bmatrix} \mathbf{b}\theta(\mathbf{v}_3) \\ \mathbf{b}a_3 \end{bmatrix} \\ \mathbf{x}_{1,r} = \mathbf{b}r, \mathbf{x}_{1,m} = \mathbf{b}\theta(\mathbf{m}_{\bar{T}}), \mathbf{x}_{1,b} = \mathbf{b} \end{cases} \quad (12)$$

As our witness has a very inhomogeneous norm, because $\mathbf{w}_1, \mathbf{w}_2$ are much larger than the rest, we use elliptic Gaussian masks as in the group signature construction of [DFJS26]. In particular, we shall note $\mathbf{x}_{1,1} = [\mathbf{x}'_{1,w,1}^T | \mathbf{x}'_{1,w,2}^T]^T$ and $\mathbf{x}_{1,2} = [\mathbf{x}'_{1,v}^T | \mathbf{x}_{1,r} | \mathbf{x}_{1,m}^T | \mathbf{x}_{1,b}]^T$, which we later refer as the first and second subwitnesses. The first has dimension $m_{1,1} = 2(dk_\pi + 1)$ while the second has dimension $m_{1,2} = (dk_H + 1)k_\pi + 1 + 1 + I'k_\pi + 1 = k_\pi(dk_H + 1 + I') + 3$. We also define the following public matrices and vectors.

$$\mathbf{A}_\theta = M_\theta(\mathbf{A}), \mathbf{D}_\theta = M_\theta(\mathbf{D}_{\bar{T}}), \mathbf{G}_\theta = \theta([q_L \mathbf{G}_H | \mathbf{0}]), \mathbf{B}_\theta = \theta([- \mathbf{B} | \mathbf{a}]), \mathbf{u}'_\theta = \theta(\mathbf{u}').$$

The protocol follows the same backbone as `Proveiss` but is specified for this relation. Notably, we split the mask and masked opening $\mathbf{y}_1, \mathbf{z}_1$ in two subvectors for the two subwitnesses. This leads to parameters that tightly match the constraint of the given subvector and only slightly changes the final rejection step.

Algorithm A.8: Prove.Rd₁($\mathbf{x}_1$)

1. $\mathbf{x}_{2,1} \leftarrow \mathcal{B}_1^{m_2-d_\pi}, \mathbf{x}_{2,2} \leftarrow \mathcal{B}_2^{d_\pi}$
2. $\mathbf{y}_{1,1} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_{1,1}, \sigma_{1,1}}}, \mathbf{y}_{1,2} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_{1,2}, \sigma_{1,2}}}, \mathbf{y}_1 \leftarrow [\mathbf{y}_{1,1}^T | \mathbf{y}_{1,2}^T]^T$
3. $\mathbf{y}_{2,1} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{m_2-d_\pi, \sigma_2}}, \mathbf{y}_{2,2} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{d_\pi, \sigma_2}}$
4. $\mathbf{y}_{3,1} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{128/n_\pi, \sigma_{3,1}}}, \mathbf{y}_{3,2} \leftarrow \mathcal{D}_{\mathcal{R}_\pi^{128/n_\pi, \sigma_{3,2}}}$
5. $\mathbf{g} \leftarrow \text{proj}(\mathbf{B}_g \mathbf{x}_{2,1} \bmod q_\pi \mathcal{R}_\pi) \triangleright \mathbf{g} \in \{a \in \mathcal{R}_{\pi, q_\pi} : \tau_0(a) = \tau_{n_\pi/2}(a) = 0\}^\ell$
6. $\mathbf{t}_A \leftarrow \mathbf{A}_{1,1} \mathbf{x}_{1,1} + \mathbf{A}_{1,2} \mathbf{x}_{1,2} + \mathbf{A}'_2 \mathbf{x}_{2,1} + \mathbf{x}_{2,2} \bmod q_\pi \mathcal{R}_\pi$
7. $\mathbf{w} \leftarrow \mathbf{A}_{1,1} \mathbf{y}_{1,1} + \mathbf{A}_{1,2} \mathbf{y}_{1,2} + \mathbf{A}'_2 \mathbf{y}_{2,1} + \mathbf{y}_{2,2} \bmod q_\pi \mathcal{R}_\pi$
8. $\mathbf{t}_{B,y1} \leftarrow \mathbf{B}_{y1} \mathbf{x}_{2,1} + \mathbf{y}_{3,1} \bmod q_\pi \mathcal{R}_\pi$
9. $\mathbf{t}_{B,y2} \leftarrow \mathbf{B}_{y2} \mathbf{x}_{2,1} + \mathbf{y}_{3,2} \bmod q_\pi \mathcal{R}_\pi$
10. $\mathbf{t}_{B,g} \leftarrow \mathbf{B}_g \mathbf{x}_{2,1} + \mathbf{g} \bmod q_\pi \mathcal{R}_\pi$
11. $(\mathbf{w}_H, \mathbf{w}_L) \leftarrow \text{Decompose}_{q_\pi}(\mathbf{w}, \gamma)$
12. $(\mathbf{t}_{A,H}, \mathbf{t}_{A,L}) \leftarrow \text{Power2Round}_{q_\pi}(\mathbf{t}_A, D)$

Output: $(\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}, \mathbf{g}, \mathbf{t}_{A,H}, \mathbf{t}_{A,L}, \mathbf{w}_H, \mathbf{w}_L, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g})$

Algorithm A.9: Prove.Rd₂($\mathbf{x}_1, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \mathbf{R}_1, \mathbf{R}_2$)

1. $\tilde{\mathbf{z}}_{3,1} \leftarrow \tau(\mathbf{y}_{3,1}) + \mathbf{R}_1 \cdot \tau(\mathbf{x}_1)$
2. $\mathbf{x}_{1,j} \leftarrow q^{-1}(\mathbf{x}_{1,b} \mathbf{x}_{1,w,1} + \mathbf{A}_\theta \mathbf{x}_{1,b} \mathbf{x}_{1,w,2} + \mathbf{x}_{1,r} \mathbf{G}_\theta \mathbf{x}_{1,v} + \mathbf{B}_\theta \mathbf{x}_{1,b} \mathbf{x}_{1,v} - \mathbf{D}_\theta \mathbf{x}_{1,b} \mathbf{x}_{1,m} - \mathbf{u}'_\theta)$
3. $\tilde{\mathbf{z}}_{3,2} \leftarrow \tau(\mathbf{y}_{3,2}) + \mathbf{R}_2 \cdot \tau(\mathbf{x}_{1,j})$
4. $u_3 \leftarrow U([0, 1])$
5. $\text{rej}_3 \leftarrow \left\lfloor u_3 > \frac{1}{M_{3,2}} \exp \left(\frac{-2 \langle \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_2 \tau(\mathbf{x}_{1,j}) \rangle + \|\mathbf{R}_2 \tau(\mathbf{x}_{1,j})\|_2^2}{\sigma_{3,2}^2} \right) \right\rfloor$

Output: $(\text{rej}_3, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{x}_{1,j})$

Algorithm A.10: Prove.Rd₃($\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}$)

1. **for** $i \in [2\ell]$ **do**
2. $\mathbf{tmp}_i \leftarrow \sum_{j \in [128]} \gamma_{i,j} (\mathbf{e}_{j, \mathcal{R}_\pi}^* \mathbf{y}_{3,1} + \mathbf{r}_{1,j}^* \mathbf{x}_1 - \tilde{\mathbf{z}}_{3,1}[j]) + \sum_{j \in [128]} \gamma_{i,128+j} (\mathbf{e}_{j, \mathcal{R}_\pi}^* \mathbf{y}_{3,2} + \mathbf{r}_{2,j}^* \mathbf{x}_{1,j} - \tilde{\mathbf{z}}_{3,2}[j]) + \gamma_{i,257} (\mathbf{x}'_{1,w,1} \mathbf{x}_{1,w,1} - B_{w1}^2) + \gamma_{i,258} (\mathbf{x}'_{1,w,2} \mathbf{x}_{1,w,2} - B_{w2}^2) + \gamma_{i,259} (\mathbf{x}'_{1,v} \mathbf{x}_{1,v} - B_3^2) + \gamma_{i,260} (\mathbf{x}_{1,r}^* \mathbf{x}_{1,r} - \omega) + \gamma_{i,261} \mathbf{x}_{1,r}^* (\mathbf{x}_{1,r} - \mathbf{x}_{1,b} \mathbf{1}) + \gamma_{i,262} \mathbf{x}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + \sum_{j \in [1, n_\pi - 1]} \gamma_{i,262+j} \mathbf{x}_{1,b} x^{n_\pi - j}$
3. **for** $i \in [\ell]$ **do**
4. $f_i \leftarrow \mathbf{g}[i] + 2^{-1}(\mathbf{tmp}_{2i-1} + \mathbf{tmp}_{2i-1}^*) + x^{n_\pi/2} \cdot 2^{-1}(\mathbf{tmp}_{2i} + \mathbf{tmp}_{2i}^*) \bmod q_\pi \mathcal{R}_\pi$

Output: $(f_1, \dots, f_\ell)$

Recall that $\mathbf{e}_{j, \mathcal{R}_\pi} = \tau^{-1}(\mathbf{e}_j)$ where $\mathbf{e}_j$ is the j -th canonical vector with a 1 at position j and 0 elsewhere. In particular, $\mathbf{e}_{j, \mathcal{R}_\pi} \in \mathcal{R}_{\pi}^{128/n_\pi}$. The vectors $\mathbf{r}_{l,j}$ in the first two sums ($l = 1, 2$) are defined as $\mathbf{r}_{l,j} = \tau^{-1}(\mathbf{R}_l^T \mathbf{e}_j)$, meaning $\tau(\mathbf{r}_{1,j}) \in \{-1, 0, 1\}^{n_\pi m_1}$ corresponds to the j -th row of $\mathbf{R}_1$ and $\tau(\mathbf{r}_{2,j}) \in \{-1, 0, 1\}^{n_\pi \cdot dk_\pi}$ corresponds to the j -th row of $\mathbf{R}_2$. Then, by developping the expression of f_i , we can express it as follows:

$$\begin{aligned}
f_i = & \mathbf{g}[i] \\
& + 2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{x}_{1,j} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{x}_{1,j}^* \\
& + (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) \mathbf{x}'_{1,w,1} \mathbf{x}_{1,w,1}^* \\
& + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) \mathbf{x}'_{1,w,2} \mathbf{x}_{1,w,2}^* \\
& + (\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259}) \mathbf{x}'_{1,v} \mathbf{x}_{1,v}^* \\
& + (\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260}) \mathbf{x}_{1,r}^* \mathbf{x}_{1,r} \\
& + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261}) (\mathbf{x}_{1,r}^* (\mathbf{x}_{1,r} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,r}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,r}) \\
& + 2^{-1}(\gamma_{2i-1,262} + x^{n_\pi/2} \gamma_{2i,262}) (\mathbf{x}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,m}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,m}) \\
& + 2^{-1}(SR_{1,2i-1}^* + x^{n_\pi/2} SR_{1,2i}^*) \mathbf{x}_1 + 2^{-1}(SR_{1,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{x}_1^* \\
& + 2^{-1}(SE_{1,2i-1}^* + x^{n_\pi/2} SE_{1,2i}^*) \mathbf{y}_{3,1} + 2^{-1}(SE_{1,2i-1} + x^{n_\pi/2} SE_{1,2i}) \mathbf{y}_{3,1}^* \\
& + 2^{-1}(SE_{2,2i-1}^* + x^{n_\pi/2} SE_{2,2i}^*) \mathbf{y}_{3,2} + 2^{-1}(SE_{2,2i-1} + x^{n_\pi/2} SE_{2,2i}) \mathbf{y}_{3,2}^* \\
& + c_i \mathbf{x}_{1,b} + c'_i \mathbf{x}_{1,b}^* \\
& - d_i
\end{aligned}$$

where for $i^* \in [2\ell]$,

$$\begin{aligned}
SE_{1,i^*} &= \sum_{j \in [128]} \gamma_{i^*,j} \mathbf{e}_j \in \mathcal{R}_{\pi, q_\pi}^{128/n_\pi}, \quad SR_{1,i^*} = \sum_{j \in [128]} \gamma_{i^*,j} \mathbf{r}_{1,j} \in \mathcal{R}_{\pi, q_\pi}^{m_1}, \\
SE_{2,i^*} &= \sum_{j \in [128]} \gamma_{i^*,128+j} \mathbf{e}_j \in \mathcal{R}_{\pi, q_\pi}^{128/n_\pi}, \quad SR_{2,i^*} = \sum_{j \in [128]} \gamma_{i^*,128+j} \mathbf{r}_{2,j} \in \mathcal{R}_{\pi, q_\pi}^{2dk_\pi},
\end{aligned} \tag{13}$$

and the polynomials c_i, c'_i and d_i for $i \in [\ell]$ are defined by

$$c_i = 2^{-1} \left(-\gamma_{2i,262+n_\pi/2} + \sum_{j=1}^{n_\pi/2-1} (\gamma_{2i-1,262+n_\pi-j} - \gamma_{2i,262+n_\pi/2-j}) x^j \right)$$

$$+ \gamma_{2i-1,262+n_\pi/2} x^{n_\pi/2} + \sum_{j=n_\pi/2+1}^{n_\pi-1} (\gamma_{2i-1,262+n_\pi-j} + \gamma_{2i,262+3n_\pi/2-j}) x^j \Big) \quad (14)$$

$$c'_i = 2^{-1} \left(\gamma_{2i,262+n_\pi/2} + \sum_{j=1}^{n_\pi/2-1} (\gamma_{2i,262+n_\pi/2+j} - \gamma_{2i-1,262+j}) x^j - \gamma_{2i-1,262+n_\pi/2} x^{n_\pi/2} \right. \\ \left. - \sum_{j=n_\pi/2+1}^{n_\pi-1} (\gamma_{2i-1,262+j} + \gamma_{2i,262+j-n_\pi/2}) x^j \right) \quad (15)$$

$$d_i = \left(\sum_{j \in [128]} \gamma_{2i-1,j} \tilde{\mathbf{z}}_{3,1}[j] + \sum_{j \in [128]} \gamma_{2i-1,128+j} \tilde{\mathbf{z}}_{3,2}[j] + \gamma_{2i-1,257} B_{w1}^2 + \gamma_{2i-1,258} B_{w2}^2 \right. \\ \left. + \gamma_{2i-1,259} B_3^2 + \gamma_{2i-1,260} \omega \right) \\ + x^{n_\pi/2} \left(\sum_{j \in [128]} \gamma_{2i,j} \tilde{\mathbf{z}}_{3,1}[j] + \sum_{j \in [128]} \gamma_{2i,128+j} \tilde{\mathbf{z}}_{3,2}[j] + \gamma_{2i,257} B_{w1}^2 + \gamma_{2i,258} B_{w2}^2 \right. \\ \left. + \gamma_{2i,259} B_3^2 + \gamma_{2i,260} \omega \right) \quad (16)$$

Notice that the term $\mathbf{x}_{1,j}$ is purely quadratic, except for a constant term in $q^{-1}\mathbf{u}'_\theta$. When computing the garbage terms in round 4, and in the verification in the recomputation of t_0 , we separate the purely quadratic part from the constant part.

Algorithm A.11: Prove.Rd₄($\mathbf{x}_1, \mathbf{x}_{2,1}, \mathbf{y}_1, \mathbf{y}_{2,1}; \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}, (\mu_i)_i$)

1. $\widehat{\mathbf{m}}_{y1} \leftarrow -\mathbf{B}_{y1}\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
2. $\widehat{\mathbf{m}}_{y2} \leftarrow -\mathbf{B}_{y2}\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
3. $\widehat{\mathbf{m}}_g \leftarrow -\mathbf{B}_g\mathbf{y}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
4. Compute $SE_{1,i}, SE_{2,i}, SR_{1,i}, SR_{2,i}$ for $i \in [2\ell]$ and c_i, c'_i for $i \in [\ell]$ as in Equations (13), (14), (15).
5. $\mathbf{y}_{1,j}^{(\text{quad})} \leftarrow (q^{-1} \bmod q_\pi)(\mathbf{y}_{1,b}\mathbf{y}_{1,w,1} + \mathbf{A}_\theta\mathbf{y}_{1,b}\mathbf{y}_{1,w,2} + \mathbf{y}_{1,r}\mathbf{G}_\theta\mathbf{y}_{1,v} + \mathbf{B}_\theta\mathbf{y}_{1,b}\mathbf{y}_{1,v} - \mathbf{D}_\theta\mathbf{y}_{1,b}\mathbf{y}_{1,m}) \bmod q_\pi \mathcal{R}_\pi$
6. $e_0 \leftarrow \sum_{i \in [\ell]} \mu_i (2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2}SR_{2,2i}^*)\mathbf{y}_{1,j}^{(\text{quad})} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2}SR_{2,2i})\mathbf{y}_{1,j}^{(\text{quad})*} + (\gamma_{2i-1,257} + x^{n_\pi/2}\gamma_{2i,257})\mathbf{y}'_{1,w,1} * \mathbf{y}'_{1,w,1} + (\gamma_{2i-1,258} + x^{n_\pi/2}\gamma_{2i,258})\mathbf{y}'_{1,w,2} * \mathbf{y}'_{1,w,2} + (\gamma_{2i-1,259} + x^{n_\pi/2}\gamma_{2i,259})\mathbf{y}'_{1,v} * \mathbf{y}'_{1,v} + (\gamma_{2i-1,260} + x^{n_\pi/2}\gamma_{2i,260})\mathbf{y}_{1,r}^* \mathbf{y}_{1,r} + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2}\gamma_{2i,261})(\mathbf{y}_{1,r}^*(\mathbf{y}_{1,r} - \mathbf{y}_{1,b}\mathbf{1}) + (\mathbf{y}_{1,r}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*)\mathbf{y}_{1,r}) + 2^{-1}(\gamma_{2i-1,262} + x^{n_\pi/2}\gamma_{2i,262})(\mathbf{y}_{1,m}^*(\mathbf{y}_{1,m} - \mathbf{y}_{1,b}\mathbf{1}) + (\mathbf{y}_{1,m}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*)\mathbf{y}_{1,m})) + \mu_{\ell+1}\mathbf{y}_{1,b}^2$
7. $\widehat{\mathbf{y}}_{1,j}^{(\text{cross1})} \leftarrow (q^{-1} \bmod q_\pi)(\mathbf{y}_{1,b}\mathbf{x}_{1,w,1} + \mathbf{A}_\theta\mathbf{y}_{1,b}\mathbf{x}_{1,w,2} + \mathbf{y}_{1,r}\mathbf{G}_\theta\mathbf{x}_{1,v} + \mathbf{B}_\theta\mathbf{y}_{1,b}\mathbf{x}_{1,v} - \mathbf{D}_\theta\mathbf{y}_{1,b}\mathbf{x}_{1,m}) \bmod q_\pi \mathcal{R}_\pi$
8. $\widehat{\mathbf{x}}_{1,j}^{(\text{cross2})} \leftarrow (q^{-1} \bmod q_\pi)(\mathbf{x}_{1,b}\mathbf{y}_{1,w,1} + \mathbf{A}_\theta\mathbf{x}_{1,b}\mathbf{y}_{1,w,2} + \mathbf{x}_{1,r}\mathbf{G}_\theta\mathbf{y}_{1,v} + \mathbf{B}_\theta\mathbf{x}_{1,b}\mathbf{y}_{1,v} - \mathbf{D}_\theta\mathbf{x}_{1,b}\mathbf{y}_{1,m}) \bmod q_\pi \mathcal{R}_\pi$
9. $e_1 \leftarrow \sum_{i \in [\ell]} \mu_i (\widehat{\mathbf{m}}_g[i] + 2^{-1}(SR_{1,2i-1}^* + x^{n_\pi/2}SR_{1,2i}^*)\mathbf{y}_1 + 2^{-1}(SR_{1,2i-1} + x^{n_\pi/2}SR_{1,2i})\mathbf{y}_1^* + 2^{-1}(SE_{1,2i-1}^* + x^{n_\pi/2}SE_{1,2i}^*)\widehat{\mathbf{m}}_{y1} + 2^{-1}(SE_{1,2i-1} + x^{n_\pi/2}SE_{1,2i})\widehat{\mathbf{m}}_{y1}^* + 2^{-1}(SE_{2,2i-1}^* + x^{n_\pi/2}SE_{2,2i}^*)\widehat{\mathbf{m}}_{y2} + 2^{-1}(SE_{2,2i-1} + x^{n_\pi/2}SE_{2,2i})\widehat{\mathbf{m}}_{y2}^* + c_i\mathbf{y}_{1,b} + c'_i\mathbf{y}_{1,b}^* + 2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2}SR_{2,2i}^*)(\widehat{\mathbf{y}}_{1,j}^{(\text{cross1})} +$

$\widehat{\mathbf{xy}}_{1,j}^{(\text{cross2})}) + 2^{-1}(SR_{2,2i-1} + x^{n\pi/2}SR_{2,2i})(\widehat{\mathbf{xy}}_{1,j}^{(\text{cross1})})^* + \widehat{\mathbf{xy}}_{1,j}^{(\text{cross2})})^* + (\gamma_{2i-1,257} + x^{n\pi/2}\gamma_{2i,257})(\mathbf{y}'_{1,w,1} * \mathbf{x}'_{1,w,1} + \mathbf{x}'_{1,w,1} * \mathbf{y}'_{1,w,1}) + (\gamma_{2i-1,258} + x^{n\pi/2}\gamma_{2i,258})(\mathbf{y}'_{1,w,2} * \mathbf{x}'_{1,w,2} + \mathbf{x}'_{1,w,2} * \mathbf{y}'_{1,w,2}) + (\gamma_{2i-1,259} + x^{n\pi/2}\gamma_{2i,259})(\mathbf{y}'_{1,v} * \mathbf{x}'_{1,v} + \mathbf{x}'_{1,v} * \mathbf{y}'_{1,v}) + (\gamma_{2i-1,260} + x^{n\pi/2}\gamma_{2i,260})(\mathbf{y}_{1,r}^* \mathbf{x}_{1,r} + \mathbf{x}_{1,r}^* \mathbf{y}_{1,r}) + 2^{-1}(\gamma_{2i-1,261} + x^{n\pi/2}\gamma_{2i,261})(\mathbf{y}_{1,r}^* (\mathbf{x}_{1,r} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,r}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,r} + \mathbf{x}_{1,r}^* (\mathbf{y}_{1,r} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,r}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,r}) + 2^{-1}(\gamma_{2i-1,262} + x^{n\pi/2}\gamma_{2i,262})(\mathbf{y}_{1,m}^* (\mathbf{x}_{1,m} - \mathbf{x}_{1,b} \mathbf{1}) + (\mathbf{y}_{1,m}^* - \mathbf{y}_{1,b}^* \mathbf{1}^*) \mathbf{x}_{1,m} + \mathbf{x}_{1,m}^* (\mathbf{y}_{1,m} - \mathbf{y}_{1,b} \mathbf{1}) + (\mathbf{x}_{1,m}^* - \mathbf{x}_{1,b}^* \mathbf{1}^*) \mathbf{y}_{1,m})) + 2\mu_{\ell+1} \mathbf{x}_{1,b} \mathbf{y}_{1,b}$
 10. $t_0 \leftarrow \mathbf{b}^T \mathbf{y}_{2,1} + e_0 \bmod q_\pi \mathcal{R}_\pi$
 11. $t_1 \leftarrow \mathbf{b}^T \mathbf{x}_{2,1} + e_1 \bmod q_\pi \mathcal{R}_\pi$
Output: (t_0, t_1)

Algorithm A.12: $\text{Prove.Rd}_5 \left(\begin{array}{c} \mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_{1,1}, \mathbf{y}_{1,2}, \\ \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{t}_{A,L}, \mathbf{w}_L, \mathbf{w}_H; \mathbf{R}_1, \tilde{\mathbf{z}}_{3,1}, c \end{array} \right)$

1. $\mathbf{z}_{1,1} \leftarrow \mathbf{y}_{1,1} + c\mathbf{x}_{1,1}$ and $\mathbf{z}_{1,2} \leftarrow \mathbf{y}_{1,2} + c\mathbf{x}_{1,2}$
 2. $\mathbf{z}_{2,1} \leftarrow \mathbf{y}_{2,1} + c\mathbf{x}_{2,1}$ and $\mathbf{z}'_{2,2} \leftarrow \mathbf{y}_{2,2} + c\mathbf{x}_{2,2}$
 3. $\mathbf{z}'_2 \leftarrow \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}'_{2,2} \end{bmatrix}$, and $\mathbf{x}_2 \leftarrow \begin{bmatrix} \mathbf{x}_{2,1} \\ \mathbf{x}_{2,2} \end{bmatrix}$
 4. $u_1, u_2 \leftarrow U([0, 1])$
 5. $\text{num} \leftarrow \pi \left(\frac{\|\mathbf{cx}_{1,1}\|_2^2}{\sigma_{1,1}^2} + \frac{\|\mathbf{cx}_{1,2}\|_2^2}{\sigma_{1,2}^2} + \frac{\|\mathbf{R}_1 \tau(\mathbf{x}_1)\|_2^2}{\sigma_{3,1}^2} \right)$
 6. $\text{den} \leftarrow 2\pi \left(\frac{\langle \tau(\mathbf{z}_{1,1}), \tau(\mathbf{cx}_{1,1}) \rangle}{\sigma_{1,1}^2} + \frac{\langle \tau(\mathbf{z}_{1,2}), \tau(\mathbf{cx}_{1,2}) \rangle}{\sigma_{1,2}^2} + \frac{\langle \tilde{\mathbf{z}}_{3,1}, \mathbf{R}_1 \tau(\mathbf{x}_1) \rangle}{\sigma_{3,1}^2} \right)$
 7. $\text{rej}_1 \leftarrow \llbracket u_1 > \frac{\exp(\text{num})}{M_{1,1} M_{1,2} M_{3,1} \cosh(\text{den})} \rrbracket$
 8. $\text{rej}_2 \leftarrow \llbracket u_2 > \frac{\exp(\pi \|\mathbf{cx}_2\|_2^2 / \sigma_2^2)}{M_2 \cosh(2\pi \langle \tau(\mathbf{z}'_2), \tau(\mathbf{cx}_2) \rangle / \sigma_2^2)} \rrbracket$
 9. $\mathbf{z}_{2,2} \leftarrow \mathbf{z}'_{2,2} - c\mathbf{t}_{A,L} - \mathbf{w}_L$
 10. $\text{rej}_{\text{comp}} \leftarrow \llbracket \|\mathbf{z}_{2,1}\|_2^2 + \|\mathbf{z}_{2,2}\|_2^2 > B_{\pi,2} \rrbracket$
 11. $\mathbf{h} \leftarrow \text{MakeGHint}_{q_\pi}(\mathbf{z}_{2,2}, \gamma \mathbf{w}_H - \mathbf{z}_{2,2}, \gamma)$

Output: $(\text{rej}_1, \text{rej}_2, \text{rej}_{\text{comp}}, \mathbf{z}_{1,1}, \mathbf{z}_{1,2}, \mathbf{z}_{2,1}, \mathbf{h})$

Algorithm A.13: $\text{Prove}(\text{statement}, \mathbf{x}_1)$

1. $(\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_1, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}, \mathbf{g}, \mathbf{t}_{A,H}, \mathbf{t}_{A,L}, \mathbf{w}_H, \mathbf{w}_L, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}) \leftarrow \text{Prove.Rd}_1(\mathbf{x}_1)$
 2. $\text{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \mathbf{w}_H)$
 3. $(\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, \mathbf{R}_{2,0}, \mathbf{R}_{2,1}) \leftarrow \mathcal{H}(1, \text{statement}, \text{msg}_1) \in (\{0, 1\}^{128 \times n_\pi m_1})^2 \times (\{0, 1\}^{128 \times nd})^2$
 4. $\mathbf{R}_1 \leftarrow \mathbf{R}_{1,0} - \mathbf{R}_{1,1}, \mathbf{R}_2 \leftarrow \mathbf{R}_{2,0} - \mathbf{R}_{2,1}$
 5. $(\text{rej}_3, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{x}_{1,j}) \leftarrow \text{Prove.iss.Rd}_2(\mathbf{x}_1, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \mathbf{R}_1, \mathbf{R}_2)$
 6. **if** $\llbracket \text{rej}_3 = 1 \rrbracket$ **goto** 1.
 7. $\text{msg}_2 = (\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2})$
 8. $(\gamma_{i,j})_{\substack{i \in [2\ell] \\ j \in [261+n_\pi]}} \leftarrow \mathcal{H}(2, \text{statement}, \text{msg}_1, \text{msg}_2) \in \mathbb{Z}_{q_\pi}^{2\ell \times (261+n_\pi)}$
 9. $(f_1, \dots, f_\ell) \leftarrow \text{Prove.Rd}_3(\mathbf{x}_1, \mathbf{x}_{1,j}, \mathbf{y}_{3,1}, \mathbf{y}_{3,2}; \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j})$
 10. $\text{msg}_3 = (f_1, \dots, f_\ell)$
 11. $(\mu_i)_{i \in [\ell+1]} \leftarrow \mathcal{H}(3, \text{statement}, \text{msg}_1, \text{msg}_2, \text{msg}_3) \in \mathcal{R}_{\pi, q_\pi}^{\ell+1}$
 12. $(t_0, t_1) \leftarrow \text{Prove.Rd}_4(\mathbf{x}_1, \mathbf{x}_{2,1}, \mathbf{y}_1, \mathbf{y}_{2,1}; \mathbf{R}_1, \mathbf{R}_2, (\gamma_{i,j})_{i,j}, (\mu_i)_i)$
 13. $\text{msg}_4 = (t_0, t_1)$
 14. $c \leftarrow \mathcal{H}(4, \text{statement}, \text{msg}_1, \text{msg}_2, \text{msg}_3, \text{msg}_4) \in \mathcal{C}$
 15. $(\text{rej}_1, \text{rej}_2, \text{rej}_{\text{comp}}, \mathbf{z}_{1,1}, \mathbf{z}_{1,2}, \mathbf{z}_{2,1}, \mathbf{h}) \leftarrow \text{Prove.Rd}_5(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{y}_{1,1}, \mathbf{y}_{1,2}, \mathbf{y}_{2,1}, \mathbf{y}_{2,2}, \mathbf{t}_{A,L}, \mathbf{w}_L, \mathbf{w}_H; \mathbf{R}_1, \tilde{\mathbf{z}}_{3,1}, c)$
 16. **if** $\llbracket \text{rej}_1 = 1 \rrbracket \vee \llbracket \text{rej}_2 = 1 \rrbracket \vee \llbracket \text{rej}_{\text{comp}} = 1 \rrbracket$ **goto** 1.

Output: $\pi = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, f_1, \dots, f_\ell, t_1, c, \mathbf{z}_{1,1}, \mathbf{z}_{1,2}, \mathbf{z}_{2,1}, \mathbf{h})$

A.4.8 Verification. As in Section A.3.8, the verification algorithm recomputes $\mathbf{w}_H$ and t_0 and essentially checks the final challenge c is coherent. It also

checks the masked openings $\mathbf{z}_{1,1}, \mathbf{z}_{1,2}, \mathbf{z}_2$ (recovered with the hints), $\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}$ verify the desired bound. We define the following bounds, $B_{\pi,2}$ being already used in [Prove.Rd₅](#).

$$\begin{cases} B_{\pi,1,1} = c_{n_\pi m_{1,1}} \sigma_{1,1} \sqrt{n_\pi m_{1,1}} \\ B_{\pi,1,2} = c_{n_\pi m_{1,2}} \sigma_{1,2} \sqrt{n_\pi m_{1,2}} \\ B_{\pi,2} = c_{n_\pi m_2} \sigma_2 \sqrt{n_\pi m_2} + (\tilde{\eta} 2^{D-1} + \frac{\gamma}{2}) \sqrt{n_\pi d_\pi} \\ B_{z3,1} = c_{128} \sigma_{3,1} \sqrt{128} \\ B_{z3,2} = c_{128} \sigma_{3,2} \sqrt{128} \end{cases}$$

Algorithm A.14: Verify(statement, π)

1. Parse $\pi = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2}, f_1, \dots, f_\ell, t_1, c, \mathbf{z}_{1,1}, \mathbf{z}_{1,2}, \mathbf{z}_{2,1}, \mathbf{h})$
2. $\mathbf{tmp} \leftarrow \mathbf{A}_{1,1} \mathbf{z}_{1,1} + \mathbf{A}_{1,2} \mathbf{z}_{1,2} + \mathbf{A}'_2 \mathbf{z}_{2,1} - c 2^D \mathbf{t}_{A,H} \bmod q_\pi \mathcal{R}_\pi$
3. $\mathbf{w}_H \leftarrow \text{UseGHint}_{q_\pi}(\mathbf{h}, \mathbf{tmp}, \gamma)$
4. $\mathbf{z}_{2,2} \leftarrow \gamma \mathbf{w}_H - \mathbf{tmp}$
5. $\mathbf{z}_2 \leftarrow \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}_{2,2} \end{bmatrix}$
6. $\mathbf{msg}_1 = (\mathbf{t}_{A,H}, \mathbf{t}_{B,y1}, \mathbf{t}_{B,y2}, \mathbf{t}_{B,g}, \mathbf{w}_H)$
7. $\mathbf{msg}_2 = (\tilde{\mathbf{z}}_{3,1}, \tilde{\mathbf{z}}_{3,2})$
8. $\mathbf{msg}_3 = (f_1, \dots, f_\ell)$
9. $(\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, \mathbf{R}_{2,0}, \mathbf{R}_{2,1}) \leftarrow \mathcal{H}(1, \text{statement}, \mathbf{msg}_1)$
10. $(\mathbf{R}_1, \mathbf{R}_2) \leftarrow (\mathbf{R}_{1,0} - \mathbf{R}_{1,1}, \mathbf{R}_{2,0} - \mathbf{R}_{2,1})$
11. $(\gamma_{i,j})_{\substack{i \in [2\ell] \\ j \in [260+n_\pi]}} \leftarrow \mathcal{H}(2, \text{statement}, \mathbf{msg}_1, \mathbf{msg}_2)$
12. $(\mu_i)_{i \in [\ell+1]} \leftarrow \mathcal{H}(3, \text{statement}, \mathbf{msg}_1, \mathbf{msg}_2, \mathbf{msg}_3)$
13. $\widehat{\mathbf{m}}_{zy1} \leftarrow \mathbf{ct}_{B,y1} - \mathbf{B}_{y1} \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
14. $\widehat{\mathbf{m}}_{zy2} \leftarrow \mathbf{ct}_{B,y2} - \mathbf{B}_{y2} \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
15. $\widehat{\mathbf{m}}_{zg} \leftarrow \mathbf{ct}_{B,g} - \mathbf{B}_g \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
16. Compute $SE_{1,i}, SE_{2,i}, SR_{1,i}, SR_{2,i}$ for $i \in [2\ell]$ and c_i, c'_i, d_i for $i \in [\ell]$ as in Equations (13), (14), (15), (16)
17. $\mathbf{z}_{1,j}^{(\text{quad})} \leftarrow (q^{-1} \bmod q_\pi)(\mathbf{z}_{1,b} \mathbf{z}_{1,w,1} + \mathbf{A}_\theta \mathbf{z}_{1,b} \mathbf{z}_{1,w,2} + \mathbf{z}_{1,r} \mathbf{G}_\theta \mathbf{z}_{1,v} + \mathbf{B}_\theta \mathbf{z}_{1,b} \mathbf{z}_{1,v} - \mathbf{D}_\theta \mathbf{z}_{1,b} \mathbf{z}_{1,m}) \bmod q_\pi \mathcal{R}_\pi$
18. $\mathbf{z}_{1,j}^{(\text{const})} \leftarrow -(q^{-1} \bmod q_\pi) \mathbf{u}_\theta \bmod q_\pi \mathcal{R}_\pi$
19. $t_0 \leftarrow \sum_{i \in [\ell]} \mu_i (2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{z}_{1,j}^{(\text{quad})} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{z}_{1,j}^{(\text{quad})*} + (\gamma_{2i-1,257} + x^{n_\pi/2} \gamma_{2i,257}) \mathbf{z}'_{1,w,1} * \mathbf{z}'_{1,w,1} + (\gamma_{2i-1,258} + x^{n_\pi/2} \gamma_{2i,258}) \mathbf{z}'_{1,w,2} * \mathbf{z}'_{1,w,2} + (\gamma_{2i-1,259} + x^{n_\pi/2} \gamma_{2i,259}) \mathbf{z}'_{1,v} * \mathbf{z}'_{1,v} + (\gamma_{2i-1,260} + x^{n_\pi/2} \gamma_{2i,260}) \mathbf{z}_{1,r}^* \mathbf{z}_{1,r} + 2^{-1}(\gamma_{2i-1,261} + x^{n_\pi/2} \gamma_{2i,261}) (\mathbf{z}_{1,r}^* (\mathbf{z}_{1,r} - \mathbf{z}_{1,b} \mathbf{1}) + (\mathbf{z}_{1,r}^* - \mathbf{z}_{1,b}^* \mathbf{1}^*) \mathbf{z}_{1,r}) + 2^{-1}(\gamma_{2i-1,262} + x^{n_\pi/2} \gamma_{2i,262}) (\mathbf{z}_{1,m}^* (\mathbf{z}_{1,m} - \mathbf{z}_{1,b} \mathbf{1}) + (\mathbf{z}_{1,m}^* - \mathbf{z}_{1,b}^* \mathbf{1}^*) \mathbf{z}_{1,m})) + \mu_{\ell+1} \mathbf{z}_{1,b}^2 + c \cdot (\sum_{i \in [\ell]} \mu_i (\widehat{\mathbf{m}}_{zg}[i] + 2^{-1}(SR_{1,2i-1}^* + x^{n_\pi/2} SR_{1,2i}^*) \mathbf{z}_1 + 2^{-1}(SR_{1,2i-1} + x^{n_\pi/2} SR_{1,2i}) \mathbf{z}_1^* + 2^{-1}(SE_{1,2i-1}^* + x^{n_\pi/2} SE_{1,2i}^*) \widehat{\mathbf{m}}_{zy1} + 2^{-1}(SE_{1,2i-1} + x^{n_\pi/2} SE_{1,2i}) \widehat{\mathbf{m}}_{zy1}^* + 2^{-1}(SE_{2,2i-1}^* + x^{n_\pi/2} SE_{2,2i}^*) \widehat{\mathbf{m}}_{zy2} + 2^{-1}(SE_{2,2i-1} + x^{n_\pi/2} SE_{2,2i}) \widehat{\mathbf{m}}_{zy2}^* + c_i \mathbf{z}_{1,b} + c'_i \mathbf{z}_{1,b}^*)) - c^2 \cdot (\sum_{i \in [\ell]} \mu_i (d_i + f_i - 2^{-1}(SR_{2,2i-1}^* + x^{n_\pi/2} SR_{2,2i}^*) \mathbf{z}_{1,j}^{(\text{const})} + 2^{-1}(SR_{2,2i-1} + x^{n_\pi/2} SR_{2,2i}) \mathbf{z}_{1,j}^{(\text{const})*}) + \mu_{\ell+1}) - c \cdot t_1 + \mathbf{b}^T \mathbf{z}_{2,1} \bmod q_\pi \mathcal{R}_\pi$
20. $\mathbf{msg}_4 \leftarrow (t_0, t_1)$
21. $c' \leftarrow \mathcal{H}(4, \text{statement}_{\text{iss}}, \mathbf{msg}_1, \mathbf{msg}_2, \mathbf{msg}_3, \mathbf{msg}_4)$
22. $b \leftarrow \llbracket c = c' \rrbracket \wedge \llbracket \|\mathbf{z}_{1,1}\|_2 \leq B_{\pi,1,1} \rrbracket \wedge \llbracket \|\mathbf{z}_{1,2}\|_2 \leq B_{\pi,1,2} \rrbracket \wedge \llbracket \|\mathbf{z}_2\|_2 \leq B_{\pi,2} \rrbracket \wedge \llbracket \|\tilde{\mathbf{z}}_{3,2}\|_2 \leq B_{z3,1} \rrbracket \wedge \llbracket \|\tilde{\mathbf{z}}_{3,2}\|_2 \leq B_{z3,2} \rrbracket \wedge \llbracket \|\mathbf{h}\|_\infty \leq \frac{q_\pi - 1}{2\gamma} \rrbracket \wedge \llbracket \forall i \in [\ell], \tau_0(f_i) = \tau_{n_\pi/2}(f_i) = 0 \rrbracket$

Output: b (1 if valid, 0 otherwise)

A.4.9 Security. We now give the formal security result for the proof system (Prove, Verify) based on the framework (and security results) from [LNP22, LN22]. The latter are slightly adapted to embark our optimization of Section 4.

Lemma A.2 (Security of Show Proof). *Let $\rho = \lfloor (2^{2(\lambda+1)/n_\pi} - 1)/2 \rfloor$, and $\tilde{\eta} \in \mathbb{N}^\times$ be defining the challenge space $\mathcal{C}$ as*

$$\mathcal{C} = \left\{ c \in \mathcal{R}_\pi : c^* = c \quad \wedge \quad \|c\|_\infty \leq \rho \quad \wedge \quad \|c^{64}\|_1^{1/64} \leq \tilde{\eta} \right\}.$$

Let $\varepsilon_{3,2} > 0$, and $M_{1,1}, M_{1,2}, M_2, M_{3,1}, M_{3,2}$ in $(1, \infty)$ be rejection rates. We then define $\alpha_{3,2} = \sqrt{\pi}(\sqrt{\ln(1/\varepsilon_{3,2})} + \ln(M_{3,2}) + \sqrt{\ln(1/\varepsilon_{3,2})})/\ln(M_{3,2})$ (for unimodal rejection sampling), and $\alpha_X = \sqrt{\pi/\ln M_X}$ for $X \in \{(1,1), (1,2), 2, (3,1)\}$ be the associated rejection sampling slacks. We define the witness $\mathbf{x}_1$ and subwitness $\mathbf{x}_{1,1}, \mathbf{x}_{1,2}$ as in Equation (12). The subwitness are of respective dimensions $m_{1,1} = 2(dk_\pi + 1)$ and $m_{1,2} = k_\pi(dk_H + 1 + I') + 3$. We let $B_{x11} = \sqrt{B_{w1}^2 + B_{w2}^2}$ and $B_{x12} = \sqrt{B_3^2 + \omega + nI' + 1}$ be a bound the Euclidean norm of each subwitness. We then define $B = \sqrt{B_{x11}^2 + B_{x12}^2}$ to be a bound on the entire witness. We also define the lifting bound

$$B_j = \frac{B_{w1}}{q} + \frac{nd}{2}B_{w2} + \omega\sqrt{\frac{1 - 1/q_H^2}{b_H^2 - 1}}B_3 + \frac{n\sqrt{d(dk_H + 1)}}{2}B_3 + \frac{nI'\sqrt{nd}}{2} + \frac{\sqrt{nd}}{2}$$

necessary when lifting the mod- q modulo q_π . We then define the Gaussian widths of the masked openings by $\sigma_{1,1} = \alpha_{1,1}\tilde{\eta}B_{x11}$, $\sigma_{1,2} = \alpha_{1,2}\tilde{\eta}B_{x12}$, $\sigma_2 = \alpha_{2,\tilde{\eta}}\sqrt{n_\pi m_2}$ for $m_2 \in \mathbb{N}^\times$, $\sigma_{3,1} = \alpha_{3,1}\sqrt{\beta(128, 128)}B$, and $\sigma_{3,2} = \alpha_{3,2}\sqrt{\beta(128, 128)}B_j$ where $\beta(\cdot, \cdot)$ is the function defined in Lemma 4.1. We also define approximate range proof bounds $B_{arp,1} = \alpha_{3,1}C_2(128, 128)B$ and $B_{arp,2} = \alpha_{3,2}C_2(128, 128)B_j$ where $C_2(\cdot, \cdot)$ is defined in Table 4.1. Then, let d_π, ℓ, q_π be in $\mathbb{N}^\times$ such that q_π is prime with $q_\pi = 5 \bmod 8$, $q_\pi = 1 \bmod q$ and satisfies the conditions of Equation (Cond_{show}), and $\ell = \lceil \lambda/2 \log_2 q_\pi \rceil$. For the Dilithium compression, we also let γ be an even divisor of $q_\pi - 1$, and $D = \max(0, \lfloor 1 + \log_2(\gamma/(\rho n_\pi)) \rfloor)$ so that $2^{D-1}\rho n_\pi \leq \gamma$. Then, (Prove, Verify) is knowledge sound with an expected-polynomial-time extractor with soundness error

$$\varepsilon_{snd} = \frac{2}{|\mathcal{C}|} + q_\pi^{-n_\pi/2} + q_\pi^{-2\ell} + 2^{-\lambda} + \varepsilon_{\text{M-SIS}}^{(snd)}$$

and zero-knowledge with loss $\varepsilon_{zk} = \varepsilon_{\text{M-LWE}}^{(zk)} + \varepsilon_{3,2}/M_{3,2} + \text{negl}(\lambda)$. The loss $\varepsilon_{\text{M-SIS}}^{(snd)}$ is the hardness bound of M-SIS $_{n_\pi, d_\pi, m_{1,1}+m_{1,2}+m_2, q_\pi, \beta}$ where β is defined as

$$4\tilde{\eta}\sqrt{4c_{n_\pi m_{1,1}}^2\sigma_{1,1}^2n_\pi m_{1,1} + 4c_{n_\pi m_{1,2}}^2\sigma_{1,2}^2n_\pi m_{1,2} + (2c_{n_\pi m_2}\sigma_2\sqrt{n_\pi m_2} + (2^D\tilde{\eta} + \gamma)\sqrt{n_\pi d_\pi})^2},$$

while $\varepsilon_{\text{M-LWE}}^{(zk)}$ is the hardness bound of M-LWE $_{n_\pi, m_2-d_\pi-256/n_\pi-\ell-1, m_2, q_\pi, B_1, B_1}$.

A.4.10 Proof Size. We estimate the size of the zero-knowledge proof π produced by [Prove](#) as in Section [A.3.10](#). We simply note that the masked opening $\mathbf{z}_1$ is now composed of two subvectors $\mathbf{z}_{1,1}, \mathbf{z}_{1,2}$ that are Gaussians with different parameters, which then leads to two separate terms. Otherwise, the estimation strategy remains the same. All the parameters are set according to Lemma [A.2](#), which leads to

$$\begin{aligned} |\pi| &= n_\pi d_\pi (\lceil \log_2 q_\pi \rceil - D + 2.25) + (256 + 2\ell + \ell n_\pi + n_\pi) \lceil \log_2 q_\pi \rceil \\ &\quad + n_\pi \lceil \log_2 (2\rho + 1) \rceil + n_\pi m_{1,1} \left(\frac{1}{2} + \log_2 \sigma_{1,1} \right) + n_\pi m_{1,2} \left(\frac{1}{2} + \log_2 \sigma_{1,2} \right) \\ &\quad + n_\pi (m_2 - d_\pi) \left(\frac{1}{2} + \log_2 \sigma_2 \right) + 128 \left(\frac{1}{2} + \log_2 \sigma_{3,1} \right) + 128 \left(\frac{1}{2} + \log_2 \sigma_{3,2} \right). \end{aligned}$$

B Suggested Parameters

Symbol	Description	Value
Parameters of TSampler		
λ	Security parameter	128
n	Signature ring degree	256
d	Module rank	4
q	Modulus	$411853 \approx 2^{18.65}$
k_L, k_H	Low and High Gadget lengths	1, 2
b_L, b_H	Low and High Gadget bases	2437, 13
q_L, q_H	Low and High Moduli	2437, 169
ε	Smoothing loss for TSampler	2^{-40}
s_L s_H	Gadget sampling widths	8174.824 44.033
s_1 s_2 s_3	Gaussian widths in TSampler	9468.026 4776.584 76.497
ω	Hamming weight of tags	8
m	Number of attributes	10
η_r	Commitment randomness binomial parameter	2
α_1, α_2	Rejection sampling slacks (sec. proof)	6.400, 3.722
M_1, M_2	Rejection sampling repetition rates (sec. proof)	1.080, 1.255
B_1 B_2 B_3	Credential bounds in Issue	135196.305 68206.036 1579.926
B_{w1} B_{w2}	Gadget sampling widths	135265.102 68274.832
Security Estimates		
$\text{BKZ}^{(\text{com})}$	Required BKZ blocksize for M-LWE ^(com)	498
$\text{BKZ}^{(\text{pk})}$	Required BKZ blocksize for M-LWE ^(pk)	462
$\text{BKZ}^{(\text{upk})}$	Required BKZ blocksize for M-LWE ^(upk)	423
BKZ	Required BKZ blocksize for M-SIS	554
$\varepsilon_{\text{M-LWE}}^{(\text{com})}$	Hardness bound of M-LWE ^(com)	$2^{-162.05}$
$\varepsilon_{\text{M-LWE}}^{(\text{pk})}$	Hardness bound of M-LWE ^(pk)	$2^{-151.52}$
$\varepsilon_{\text{M-LWE}}^{(\text{upk})}$	Hardness bound of M-LWE ^(upk)	$2^{-140.12}$
$\varepsilon_{\text{M-SIS}}$	Hardness bound of M-SIS	$2^{-178.43}$
Efficiency Estimates		
opk	Size of issuer public key $\mathbf{B}$ (+seed)	19488 B
osk	Size of issuer secret key $\mathbf{R}_1, \mathbf{R}_2$	4096 B
upk	Size of user public key $\mathbf{b}$ (+seed)	2464 B
upk	Size of user secret key $\mathbf{s}_1, \mathbf{s}_2$	256 B
cred	Size of credential $(t, \mathbf{v}_2, \mathbf{v}_3)$	3607 B

Table B.1. Suggested parameter set for the sampler and credential generation.

Symbol	Description	Value
Proof System Parameters		
$\lambda_{snd}, \lambda_{zk}$	Security for soundness and zero-knowledge	128, 161
n_π	Proof system ring degree	64
k_π	Subring embedding dimension	4
d_π	Module rank	16
q_π	Proof system modulus	117877270837
ℓ	Soundness amplification dimension	2
m_1	Witness dimension	106
m_2	Dimension of ABDLOP randomness	55
ρ	Infinity norm of challenges	8
$\tilde{\eta}$	Manhattan-like norm of challenges	93
α_X	Gaussian rejection sampling slacks (for $X = 1, 2, (3, 1), (3, 2)$)	3.011
M_X	Rejection sampling repetition rates (for $X = 1, 2, (3, 1), (3, 2)$)	$\sqrt{2}$
σ_1	First Gaussian mask width	23949.530
σ_2	Second Gaussian mask width	16612.366
$\sigma_{3,1}$	Third Gaussian mask width (witness ARP)	3912.587
$\sigma_{3,2}$	Third Gaussian mask width (lifting ARP)	2837348.222
γ	Mask commitment compression parameter	16836
D	Witness commitment compression parameter	6
B_{x1}	Witness bound	85.534
$B_{j,1}$ $B_{j,2}$	Lifting bounds	59815.930 16400.000
$B_{arp,1}$	ARP proven bound (witness ARP)	38621.851
$B_{arp,2}$	ARP proven bound (lifting ARP)	28007973.353
$B_{h,iss,1}$	Relaxed lifting proof slack bound 1	98.066
$B_{h,iss,2}$	Relaxed lifting proof slack bound 2	97.915
Security Estimates		
$BKZ^{(zk,iss)}$	Required BKZ blocksize for M-LWE $^{(zk,iss)}$	510
$BKZ^{(snd,iss)}$	Required BKZ blocksize for M-SIS $^{(snd,iss)}$	386
$\varepsilon_{M-LWE}^{(zk,iss)}$	Hardness bound of M-LWE $^{(zk,iss)}$, Lem. A.1	$2^{-165.56}$
$\varepsilon_{M-SIS}^{(snd,iss)}$	Hardness bound of M-SIS $^{(snd,iss)}$, Lem. A.1	$2^{-129.29}$
$\varepsilon_{snd,iss}$	Soundness error	$2^{-128.02}$
$\varepsilon_{zk,iss}$	Zero-Knowledge security	$2^{-165.56}$
Efficiency Estimates		
$ \pi_{iss} $	Proof size	23.66 KB

Table B.2. Suggested parameter set for the anonymous credentials' issuance proof ($\text{Prove}_{iss}, \text{Verify}_{iss}$).

Symbol	Description	Value
Proof System Parameters		
$\lambda_{snd}, \lambda_{zk}$	Security for soundness and zero-knowledge	128, 193
n_π	Proof system ring degree	64
k_π	Subring embedding dimension	4
d_π	Module rank	21
q_π	Proof system modulus	4682043065044021
ℓ	Soundness amplification dimension	2
$m_{1,1}$	Subwitness dimensions	34
$m_{1,2}$		79
m_2	Dimension of ABDLOP randomness	79
ρ	Infinity norm of challenges	8
$\tilde{\eta}$	Manhattan-like norm of challenges	93
$\varepsilon_{3,2}$	Unimodal rejection sampling loss	2^{-193}
α_X	Gaussian rejection sampling slacks (for $X = (1, 1), (1, 2), 2, (3, 1)$)	3.011
$\alpha_{3,2}$	Unimodal Gaussian rejection sampling slack	118.381
M_X	Rejection sampling repetition rates (for $X = (1, 1), (1, 2), 2, (3, 1), (3, 2)$)	$\sqrt{2}$
$\sigma_{1,1}$	First Gaussian mask widths	42425613.264
$\sigma_{1,2}$		442608.923
σ_2	Second Gaussian mask width	19909.653
$\sigma_{3,1}$	Third Gaussian mask width (witness ARP)	6931839.324
$\sigma_{3,2}$	Third Gaussian mask width (lifting ARP)	65135031171.822
γ	Mask commitment compression parameter	86703586
D	Witness commitment compression parameter	18
B_{x11}	Subwitness bounds	151519.307
B_{x12}		1580.737
B_j	Lifting bound	36212048.493
$B_{arp,1}$	ARP proven bound (witness ARP)	68425429.610
$B_{arp,2}$	ARP proven bound (lifting ARP)	642959578606.272
B_h	Relaxed lifting proof slack bound	56.561
Security Estimates		
$BKZ^{(zk)}$	Required BKZ blocksize for M-LWE $^{(zk)}$	615
$BKZ^{(snd)}$	Required BKZ blocksize for M-SIS $^{(snd)}$	384
$\varepsilon_{M-LWE}^{(zk)}$	Hardness bound of M-LWE $^{(zk)}$, Lem. A.2	$2^{-196.27}$
$\varepsilon_{M-SIS}^{(snd)}$	Hardness bound of M-SIS $^{(snd)}$, Lem. A.2	$2^{-128.71}$
ε_{snd}	Soundness error	$2^{-127.75}$
ε_{zk}	Zero-Knowledge security	$2^{-193.30}$
Efficiency Estimates		
$ \pi $	Proof size	35.45 KB

Table B.3. Suggested parameter set for the anonymous credentials’ show proof (Prove, Verify).