A Framework for Practical Anonymous Credentials from Lattices

Jonathan Bootle
jbt@zurich.ibm.com
IBM Research Europe - Zurich

Vadim Lyubashevsky
vad@zurich.ibm.com
IBM Research Europe - Zurich

Ngoc Khanh Nguyen
khanh.nguyen@epfl.ch
EPFL

Alessandro Sorniotti
aso@zurich.ibm.com
IBM Research Europe - Zurich

Abstract. We present a framework for building practical anonymous credential schemes based on the hardness of lattice problems. The running time of the prover and verifier is independent of the number of users and linear in the number of attributes. The scheme is also compact in practice, with the proofs being as small as a few dozen kilobytes for arbitrarily large (say up to 2^{128}) users with each user having several attributes. The security of our scheme is based on a new family of lattice assumptions which roughly states that given short pre-images of random elements in some set S, it is hard to create a pre-image for a fresh element in such a set. We show that if the set admits efficient zero-knowledge proofs of knowledge of a commitment to a set element and its pre-image, then this yields practically-efficient privacy-preserving primitives such as blind signatures, anonymous credentials, and group signatures. We propose a candidate instantiation of a function from this family which allows for such proofs and thus yields practical lattice-based primitives.

1 Introduction

The recent announcement that the American National Institute of Standards and Technology (NIST) will be standardizing three lattice-based encryption and digital signature schemes, together with the NSA releasing their new CNSA 2.0 crypto suite which mandates that lattice-based constructions be the main cryptographic tools for communication security beginning from 2030, it is looking increasingly likely that lattices will form the future foundation of public key cryptography. While we already have several very efficient almost-standardized schemes for encryption and digital signatures [BDK+18; DKL+18; PFH+17], the state of more advanced quantum-safe cryptography is more murky with a lot of constructions being noticeably less efficient than their classical counterparts. The last several years, however, have seen tremendous progress in constructing efficient zero-knowledge proofs for lattice relations [BLS19; YAZ+19; ESLL19; ALS20; ENS20; LNP22] and this led to rather compact and practical constructions of schemes like ring signatures, group signatures and confidential payment protocols [EZS+19; LNS21b; LNPS21; LNP22]. A set of important primitives that are currently lacking in truly efficient instantiations fall under the global umbrella of *anonymous credentials*.

At a very high level, in an anonymous credential scheme [Cha85; Che95; Dam88; LRSW99] an issuer signs a set of credentials for a user. The user can then present a subset of this credential set and give a zero-knowledge proof that they were indeed signed by the issuer. For the scheme to be secure, it should be impossible for a user to present a credential that the issuer never signed. For anonymity, when a user presents his credential, all his other credentials should remain hidden – even from the issuer. Furthermore, multiple credential presentations by the same user should remain unlinkable. Manifestations of such schemes have been adopted by the *self-sovereign identity* community, which has developed open-source projects [Hypb; Hypa; Ver; MATb] and standard drafts [LKWL22; The22] catering for a market whose size is estimated in the billions of US dollars [NFC22]. While there are some very efficient instantiations of these schemes based on non-quantum-safe assumptions, to the best of our knowledge there are no truly practical quantum-safe instantiations that are available.[1]

[1] It is of course possible to construct these schemes using succinct zero-knowledge proofs such as STARKs, but the cost of these constructions appears to be in the hundreds of kilobytes range.

Some recent lattice-based constructions of related primitives that previously didn't have any efficient quantum-safe instantiation based the security of the scheme on novel, but very plausible, lattice assumptions. For example, the recent blind signature in [AKSY22] was based on the new "one-more-ISIS" assumption and publicly-verifiable SNARKs were recently instantiated using a new k-R-ISIS assumption in [ACL+22]. The blind signature in [AKSY22] is one of the most concretely-efficient quantum-safe construction to date, while the work in [ACL+22] is the first construction of its kind that gets within the vicinity of practicality. It is not too surprising that the range of assumptions needed to be expanded in order to construct more efficient lattice-based primitives. Analogously to the non-post-quantum setting, simple primitives like digital signatures can be efficiently based on the hardness of the basic discrete logarithm problem, while the most efficient constructions of other more advanced schemes crucially rely on much more esoteric assumptions. And so, analogously to the classical setting, it makes sense to explore other assumptions for constructing advanced lattice-based primitives as well.

In this paper we propose a new family of lattice problems some of whose members admit a practically-efficient zero-knowledge proof for proving knowledge of a solution. We then show how to apply this zero-knowledge proof to create fairly simple constructions of various efficient privacy-preserving lattice-based primitives, such as blind signatures and anonymous credentials, based on the presumed hardness of the new problem. The specific assumption we use in our paper is a particular instantiation from this family of assumptions that yields efficient zero-knowledge proofs. The family of problems is very natural and can be seen as a generalization of the underlying problem upon which the classic GPV signature scheme [GPV08] is based on.

Lattice-Based hash-and-Sign Signatures and Extensions. Privacy-based primitives are often constructed as some combination of a hash-and-sign digital signature scheme and a zero-knowledge proof system. Hash-and-sign lattice-based digital signatures based on the hardness of the standard SIS problem, first constructed in [GPV08], are abstractly based on the hardness of the following problem over some distribution of input matrices A, which is parametrized by a global function f (there are also specific parameters n, m, β, β' which are determined based on the security parameter):

Definition 1.1 (The ISIS_f Problem (informal)). *We are given a matrix $A \in \mathbb{Z}_p^{n \times m}$ (chosen from some distribution), a function $f : [N] \to \mathbb{Z}_p^n$, and access to an oracle who chooses a random input $x \in [N]$ and outputs it together with a vector $\|\vec{s}\| \leq \beta$ satisfying $A\vec{s} = f(x)$. The game is won by coming up with a fresh tuple $(x', \vec{s}') \in [N] \times \mathbb{Z}^m$ where $\|\vec{s}'\| \leq \beta'$ and $A\vec{s}' = f(x')$.*[2]

The hardness of the above problem depends on the choice of f, the distribution of A and how the oracle chooses the vector $\vec{s}$.[3] When the function f is instantiated as a cryptographic hash function (e.g. SHA) which is modelled as a random oracle, A is chosen uniformly-random, and $\vec{s}$ is chosen from a distribution with enough entropy, then the ISIS_f problem is equivalent to the well-known SIS and ISIS problems. If f is modelled as a random oracle, then $f(x)$ is uniformly-random in $\mathbb{Z}_p^n$, and so there is no advantage gained by seeing a pre-image $\vec{s}$ of a random element in $\mathbb{Z}_p^n$ because one could, in principle, do the pre-image sampling in reverse. That is, one could instead generate a random $\vec{s}$ and if the distribution of $\vec{s}$ has enough entropy, then it is the pre-image of a uniformly-random $\vec{t} = A\vec{s}$ (since $(A, \vec{t} = A\vec{s})$ is uniformly-random by the leftover hash lemma).

Modelling f as a random oracle and furthermore being able to create a matrix A together with a trapdoor which allows for pre-image sampling, gives rise to lattice-based hash-and-sign signatures based on the hardness of ISIS_f [GPV08]. The idea is simply to sign a message (digest) x with the pre-image $\vec{s}$. This allows us to sign random messages x, but when when f is modelled as a random oracle, it doesn't matter whether x is chosen at random or adaptively because $f(x)$ is uniformly-random regardless – thus allowing the signing of arbitrary message is as secure as only being able to sign random ones.

[2] Note that x' can be equal to one of the x, as long as $\vec{s}' \neq \vec{s}$.

[3] We will also want the set $[N]$ to be exponentially large so that there is a negligible chance of obtaining pre-images on the same $f(x)$.

When f is not a cryptographic hash function, then we don't immediately obtain a signature scheme based on ISIS_f by treating x as the message to be signed precisely because the ISIS_f problem requires x to be random, whereas in an attack on a signature scheme the x is chosen by the adversary. One could, of course, propose a stronger version of ISIS_f in which the x is chosen by the attacking algorithm, but we feel that this may require the function f to be needlessly "complicated" (e.g. something that's close to a cryptographic hash function like SHA) in order for the problem to remain hard.[4] And we do not want the function circuit to be too complex because building anonymous credentials from a signature scheme will additionally require proving knowledge of the signature. Even though recent techniques for proving arbitrarily large circuits have proof sizes of around 60 KB [BS22], this is still larger than proofs of simple linear or quadratic relations which can be as small as a dozen KB for common lattice relations [LNP22]. Furthermore, the more complex proofs of arbitrary circuits will be less efficient, in terms of computation time, than proofs of simpler relations.

The main result of this work is to show how one can build an anonymous credential scheme based upon the above-sketched construction of a signature scheme for random messages. We also propose a simple function f which admits a very efficient zero-knowledge proof of the relation $A\vec{s}' = f(x')$, and for which we believe that the ISIS_f problem is hard. We now discuss our choice of f and will sketch the anonymous credential scheme construction and proof in Section 1.1.

Arguably the simplest way to instantiate $f(x)$ is to let it be a linear function. For example, if we use the natural correspondence between binary vectors of length $\log N$ with the set $[N]$, then if $f(\vec{v}) = B\vec{v}$, where $\vec{v} \in \{0,1\}^{\log N}$, then one can use the practically-efficient zero-knowledge proofs from [LNP22] to commit to $\vec{s}$ and $\vec{v}$ and show that they have small norms and satisfy $A\vec{s} = B\vec{v}$. At first glance, it might seem that the linearity of the function interacts dangerously with the inherent linearity of lattices. For example, if $\vec{v}_1 + \vec{v}_2 = \vec{v}_3$ and we have $A\vec{s}_i = f(\vec{v}_i)$ for $i = 1, 2$, then $A(\vec{s}_1 + \vec{s}_2) = f(\vec{v}_3)$. In our definition of the problem, however, the $\vec{v}_i$ are chosen at random and it is very unlikely that the sum of two randomly-chosen binary vectors will sum to another binary one (i.e. the probability is $(3/4)^{\log N}$). And if we sum more than 2 binary vectors, then the probability that the resulting pre-image $\vec{s}$ which is a sum of the pre-images, will still be as small becomes negligible.[5] So the hardness of ISIS_f for the above-defined function relies in large part on the fact that the domain of f is not closed under addition, even though f is a linear function over a larger domain. And so we conjecture that instantiating f in this way (i.e. choosing f as a random linear function over some ring and its domain as binary vectors) leads to a hard instance of ISIS_f. Furthermore, proving $A\vec{s} = B\vec{v}$ for some short $\vec{s}$ and binary $\vec{v}$ can be done very efficiently using the proofs from [LNP22]. In particular, we do not see any better algorithm for solving the ISIS_f problem when f is defined as above vs. when f is a cryptographic hash function as in [GPV08].

1.1 Blind Signatures and Anonymous Credentials from ISIS_f

We will now give a high level sketch of our main construction – an efficient anonymous credential scheme based on the hardness of the ISIS_f problem when one is able to give an efficient zero-knowledge proof of a solution to ISIS_f. We will first sketch the construction and proof of a blind signature scheme and then explain how essentially the same protocol can be turned into an anonymous credential scheme.

The Blind Signature Scheme. Recall that in a blind signature scheme, the user asks a signer to sign a message $\vec{m}$ and can then prove to everyone that he has a signature for $\vec{m}$. The blindness property of the scheme states that the signer cannot know during which interaction he signed $\vec{m}$, and the soundness property states that a user who interacted with the signer k times can only produce k valid signed messages.

The signer creates a matrix A and a trapdoor which allows him to sample short $\vec{s} \sim D_\sigma$, for some standard deviation σ, satisfying $A\vec{s} = \vec{t}$ for any $\vec{t} \in \mathbb{Z}_p^n$. In the concrete instantiation, the matrix A, along with the

[4] Still, investigating this stronger assumption with appropriate functions f could be an interesting research direction. The one-more-ISIS assumption in [AKSY22] does allow the adversary access to an oracle to obtain pre-images of arbitrary vectors, but requires him to find pre-images for a random set of vectors to win the game.

[5] For example, the expected squared norm of a 512-dimensional Gaussian of standard deviation σ is $512\sigma^2$. The probability that the sum of three such Gaussians have squared norm less than this is less than 2^{-160}.

trapdoor and sampling algorithm, is as in the NTRU-based signature scheme [DLP14; PFH+17].[6] The signer also creates random matrices B_1 and B_2 (their dimensions will be clear from their usage). The public key is A, B_1, B_2 and the function f, while the secret key is the trapdoor for A.

If the user wants to sign a (digest of a) message $\vec{m}$, he sends to the signer an Ajtai commitment of his message committed under randomness $\vec{r}$ as $\vec{c} = B_1\vec{m} + B_2\vec{r}$, along with a zero-knowledge proof (using [LNP22]) that $\vec{m}$ and $\vec{r}$ have small norms and the linear relation $\vec{c} = B_1\vec{m} + B_2\vec{r}$ is satisfied. The signer checks the proof, creates a uniformly-random "tag" $x \in [N]$ and then uses the trapdoor for A to create a short pre-image $\vec{s}$ satisfying

$$A\vec{s} = f(x) + B_1\vec{m} + B_2\vec{r}. \tag{1}$$

The pre-image $\vec{s}$ and the tag x are sent to the user. In order to prove that he has a signature of a message $\vec{m}$, the user reveals $\vec{m}$ and creates a zero-knowledge proof of knowledge of $\vec{s}, x$ satisfying (1). In the case that f is a linear function, the exact same proof from [LNP22] for proving $\vec{c} = B_1\vec{m} + B_2\vec{r}$ can be used here as well.

The proof of anonymity of the blind signature (i.e. that the signer cannot figure out during which interaction $\vec{m}$ was signed) is insured by the fact that $\vec{m}$ is transferred to the signer in a computationally-hiding commitment scheme and all the proofs are zero-knowledge. The more interesting part of the proof is showing that a user cannot produce more signed messages than the number of queries that he makes to the signer. We will show that if there is an adversary who can forge in the blind signature scheme, then there is an algorithm who can solve the ISIS_f problem.

Given the matrix A and a function f in the ISIS_f problem, the reduction chooses matrices with small coefficients R_1 and R_2, and then creates $B_1 = AR_1$ and $B_2 = AR_2$. By the LWE assumption, these look indistinguishable from uniform. The public key for the blind signature is thus A, B_1, B_2, along with the function f. In his first move, the adversary sends an Ajtai commitment to a message $\vec{m}$ using randomness $\vec{r}$ along with a zero-knowledge proof that the commitment is valid. In particular, the zero-knowledge proof ensures that the commitment is of the form $B_1\vec{m} + B_2\vec{r}$, and the reduction can extract the $\vec{m}$ and $\vec{r}$. By the construction of B_1 and B_2, we have $B_1\vec{m} + B_2\vec{r} = A(R_1\vec{m} + R_2\vec{r})$. The reduction then calls the oracle in the ISIS_f definition and receives some $\vec{s}, x$ satisfying $A\vec{s} = f(x)$. Notice that the reduction can now create $\vec{s}' = \vec{s} + R_1\vec{m} + R_2\vec{r}$ that satisfies

$$A\vec{s}' = f(x) + B_1\vec{m} + B_2\vec{r}. \tag{2}$$

The tuple $(\vec{s}', x)$ could be a valid signature except for the fact that the distribution of $\vec{s}'$ is not a discrete Gaussian (as in the real scheme), but is a Gaussian perturbed by $R_1\vec{m} + R_2\vec{r}$. To get $\vec{s}'$ to be a discrete Gaussian, the reduction can use rejection sampling as in [Lyu12, Theorem 4.6] which converts shifted Gaussians into zero-centered ones. If we accept, then we can send $\vec{s}'$ and x to the adversary as the signature of $\vec{m}$. If there is a rejection, then we can again query the oracle to obtain another $\vec{s}$ and x and create another potential vector $\vec{s}'$, and so on until the rejection sampling procedure accepts.

The above shows that the view of the adversary is identical in the simulation as in the real signature scheme, and thus the adversary should be successful in creating a signature for a new message $\vec{m}$. In other words, he is able to prove knowledge of an $x \in [N]$ and vectors $\vec{m}, \vec{r}$ satisfying (1), and these can be extracted by the extractor. Since $B_1 = AR_1$ and $B_2 = AR_2$, we have the potential solution to the ISIS_f problem being

$$A(\vec{s} - R_1\vec{m} - R_2\vec{r}) = f(x). \tag{3}$$

There are three possibilities for the x that is used in the forgery

1. x has not been queried to the oracle
2. x has been queried to the oracle, but was not seen by the adversary (i.e. it was discarded by the reduction during rejection sampling)
3. x has been queried to the oracle and seen by the adversary

[6] This implies that the scheme would be instantiated over some polynomial ring rather than $\mathbb{Z}_p$, as in the description in this section. But since $\mathbb{Z}_p$ is a sub-ring of the polynomial ring, all operations in the polynomial ring can also be described as operations over vectors over $\mathbb{Z}_p$, and so the description in this section is actually more general than we will need to be.

In case 1, (3) directly gives a solution to the ISIS_f problem. In case 2, the fact that x was already queried means that the reduction has a vector $\vec{s}'$ satisfying $A\vec{s}' = f(x)$. If $\vec{s}' \neq \vec{s} - R_1\vec{m} - R_2\vec{r}$, then we again have a solution to ISIS_f. Because the adversary never saw $\vec{s}'$ and the entropy of pre-image sampling is (exponentially) large, there is only a negligible chance that $\vec{s}' = \vec{s} - R_1\vec{m} - R_2\vec{r}$.

In case 3, the adversary knows the value $\tilde{s} = \vec{s}' + R_1\vec{m}' + R_2\vec{r}'$ where $\vec{s}'$ was such that $A\vec{s}' = f(x)$, and it produces a forgery $\vec{s}, \vec{m} \neq \vec{m}', \vec{r}$ satisfying $A(\vec{s} - R_1\vec{m} - R_2\vec{r}) = f(x)$. If $\vec{s}' = \tilde{s} - R_1\vec{m}' - R_2\vec{r} \neq \vec{s} - R_1\vec{m} - R_2\vec{r}$, then we have a solution to the ISIS_f problem. In order for this to not be a solution, the adversary would need to create $\vec{s}, \vec{m} \neq \vec{m}', \vec{r}$ such that

$$(\tilde{s} - \vec{s}) + R_1(\vec{m} - \vec{m}') + R_2(\vec{r} - \vec{r}') = 0. \tag{4}$$

To prove that even an all-powerful adversary will fail in creating such an equality (with non-negligible probability), we need to rely on the entropy of the matrix R_1. In particular, we can set the distribution of R_1 such that for every column of R_1, there exists another R_1' which differs from R_1 in only that column, satisfying $AR_1 = AR_1'$ (c.f. [LM18, Lemma 4.4]). If for every R_1, there is such a set of R_1', then for the R_1' which differs in the column corresponding to the coordinate where where $\vec{m}$ and $\vec{m}'$ differ, we will have $R_1(\vec{m} - \vec{m}') \neq R_1'(\vec{m} - \vec{m}')$. An important point is also that when receiving the vector $\tilde{s}$, which is a result of rejection sampling that involves $R_1\vec{m}$, the value of R_1 is hidden since the output of the rejection sampling hides the specific $R_1\vec{m}$. In short, the success probability of the reduction breaking the ISIS_f problem is at least half the probability that an adversary can forge the blind signature.

Adapting the above to anonymous credentials and other privacy primitives. The framework for creating an anonymous credential scheme is virtually identical to the one above for a blind signature scheme. The first part of the scheme is the issuance of a credential for a set of attributes $\vec{m}$. This is done in the exact same manner as signing in the blind signature scheme. The user submits a commitment to a vector of credentials $\vec{m}$ along with a zero-knowledge proof that the commitment is validly formed, and the issuer creates a random tag x along with a pre-image vector $\vec{s}$ satisfying (1). The one difference in an anonymous credential scheme over a blind signature is that the user may not wish to reveal the entire attribute vector $\vec{m}$, whereas in the blind signature scheme, the whole message is revealed. In this case, he can simply reveal the sub-vector $\vec{m}'$ of $\vec{m}$ that he wishes and then prove knowledge of the remaining part of $\vec{m}$ in the zero-knowledge proof. The security proof for the anonymous credential scheme is virtually identical to the one for the blind signature.

One can also easily adapt the framework to create a group signature scheme. The only change versus an anonymous credential scheme would be that the user would additionally create a lattice-based encryption of x along with the zero-knowledge proof, and additionally prove (again using the zero-knowledge proof from [LNP22]) that the ciphertext is a valid encryption of x. The size of the signature will be larger than that in [LN22], but the advantage is that signing and verification time does not scale linearly in the group size.

1.2 Related Work

Prior works building privacy-preserving primitives such as group signatures, blind signatures, and anonymous credentials [PLS18; LNPS21; PK22; JRLS22], circumvented the need to prove knowledge of a random oracle pre-image by using a standard-model digital signature framework of [ABB10] instead of the hash-and-sign based approach. But these standard-model signatures are a factor 5X - 10X longer because instantiating standard-model signatures requires larger parameters and does not allow instantiating the trapdoored matrix A via the NTRU assumption.

The more recent constructions of efficient blind signatures [AKSY22; BLNS23] which do not start from the signature in [ABB10], and do in fact use the NTRU trapdoor, unfortunately cannot be extended to anonymous credential schemes while utilizing the the compact proofs from [LNP22] for simple lattice relations. The technical reason is that in these blind signature schemes, the user gets the signer to blindly sign $\mathsf{H}(\mu)$ for some message μ and cryptographic hash function H. In the proof, he reveals the whole μ in the proof, and proves knowledge of a simple relation that includes $\mathsf{H}(\mu)$. Because the user will only ever give one proof that includes μ, revealing μ is perfectly fine. In an anonymous credential scheme, however, one may want to get

a set of credentials (i.e. $\mu = \mu'||\mu''$) blindly signed, and then reveal only the μ' part of it. This will require proving knowledge of a μ'' that satisfies some relation involving $\mathsf{H}(\mu'||\mu'')$, which results in longer and slower proofs.[7]

1.3 Discussion and Open Problems

The most pertinent open problem is to analyze the hardness of the ISIS_f problem when instantiated with a random linear f, or perhaps other classes of slightly higher-degree functions f which still yield efficient zero-knowledge proofs.

Another open problem is to come up with a possibly stronger assumption that would result in tighter reductions for our schemes. In particular, we feel that breaking our blind signature and anonymous credential scheme is more difficult than what the security reductions imply. For example, we need to set parameters so as to allow rejection sampling to work and for the matrix R_1 to be chosen such that with high probability there is another R_1' satisfying $AR_1 = AR_1'$.

If one looks closely at the proof sketch in Section 1.1, what we have essentially done is give a reduction from ISIS_f to an "interactive" version of ISIS_f in which the adversary has more control over the pre-images that he receives. In the ISIS_f problem, the adversary is given an oracle that outputs $\vec{s}, x$ satisfying $A\vec{s} = f(x)$, whereas in the "interactive" version, the adversary can choose $\vec{m}$ and $\vec{r}$ as in (1) and then receives a random x and $\vec{s}$ satisfying (1). His goal is then to satisfy (1) for a fresh $\vec{m}$. We have shown that if the random matrices B_1 and B_2 are constructed as AR_1 and AR_2, and other parameters are set appropriately,then using rejection sampling we are able to transform $\vec{s}, x$ that satisfy $A\vec{s} = f(x)$ into $\vec{s}', x$ that satisfy (1). Furthermore, picking an R_1 from a wide-enough distribution to guarantee the existence of another R_1' satisfying $AR_1 = AR_1'$ was required to assure that the adversary could not create a forgery that satisfied (4). It is very much possible, however, that setting the scheme parameters to allow for these particularities of the proof to go through is not really necessary for security. Since the asymptotically, the interactive ISIS_f problem is as hard as ISIS_f, it makes sense to examine the interactive version on its own with a more favorable parameter setting. That is, one could make the assumption that the interactive version of the ISIS_f problem is hard in the parameter range that does not yield a reduction from ISIS_f.[8] In Table 1, we give example parameters of instantiations of an anonymous credential scheme based on the ISIS_f assumption and parameters required if one assumes hardness of the interactive version. The scheme based on ISIS_f is noticeably more compact than the previously most efficient anonymous credential scheme [JRLS22], which has output sizes of around 650KB. But as we also see, there is a noticeable advantage in setting the scheme parameters smaller and assuming that the problem (interactive ISIS_f) is still hard. We believe that analyzing the security of this version of the problem, and possibly building other more efficient schemes based on it, is a promising research direction when it comes to building practical lattice-based schemes.

One of the main appeals of lattice cryptography, in addition to its versatility, is that its underlying operations are very fast when instantiated over polynomial rings. It is therefore quite conceivable that lattice-based constructions will be the fastest option out of all the post-quantum alternatives. Since most anonymous credential schemes involve real-time interaction (e.g. credit card usage), speed is a very important consideration in their real world deployment. Unlike the many efforts to construct efficient software for discrete log, pairing-based, and PCP/IOP-based proof systems, there has not been much concentrated effort to develop software for efficient lattice-based primitives. Part of the problem has been that there were not many lattice-based protocols which were compact enough to be considered for practical deployment. We hope that this paper provides some motivation for creating implementations of lattice-based privacy primitives that are real-world ready.

[7] The blind signature protocol of [BLNS23] does require a zero-knowledge proof involving $\mathsf{H}(\cdot)$, but this proof is only done in the intermediate interaction between the user and the signer, and so its relative inefficiency does not affect the signature size.

[8] This is similar to how the SIS and LWE problems were first introduced – solving their random instances was shown to be as hard as solving lattice problems in the worst case [Ajt96; MR07; Reg09], but now the SIS and LWE problems are used with parameters which do not satisfy these original reductions because these problems have since been very well-studied on their own.

Attributes	Assumption			
	ISIS_f	interactive ISIS_f	Strong-RSA [CL02a]	qSDH [CDL16]
8	122KB	26KB	1319B	608B
16	133KB	29KB	1910B	865B

Table 1: Output sizes for the anonymous credential schemes with 8 and 16 attributes.

1.4 Paper Organization

We start by covering relevant preliminaries in Section 2. This includes the necessary background on lattices, computational assumptions and non-interactive zero-knowledge proofs in the random oracle model. Section 3 introduces a new family of computational assumptions, called ISIS_f. We further provide example instantiations of functions f, and give an overview of naive (lattice and combinatorial) attacks on ISIS_f. Additionally, we introduce an interactive version of ISIS_f, denoted as $\mathsf{Int\text{-}ISIS}_f$, where the adversary is allowed to make queries before outputting a solution. The reason why we define two separate problems is because $\mathsf{Int\text{-}ISIS}_f$ is a much more approachable problem to reduce the security of a scheme to. The main result of the main body is a security reduction from $\mathsf{Int\text{-}ISIS}_f$ to ISIS_f. Next, in Section 3.3 we provide a natural template to build exotic signatures from the (interactive) ISIS_f assumption.

Section 4 provides background on anonymous credentials, including definitions and related works. Then, in Section 5 we describe a proof of knowledge of a $\mathsf{Int\text{-}ISIS}_f$ solution for a specific function f used in our anonymous credentials. To this end, we adapt the protocol from the recent zero-knowledge framework by Lyubashevsky et al. [LNP22]. Since this work is relatively new, we provide additional background on the protocol along with the intuition. Further, Section 6 presents a zero-knowledge proof for proving knowledge of a commitment opening. For security reasons, we will require the protocol to satisfy so-called straight-line extractability. Towards this goal, we combine the [LNP22] framework with the recent Katsumata transform [Kat21], which allows us to achieve this property. Section 7 introduces a falsifiable version of the (interactive) ISIS_f problem using NTRU lattices. We show that the main reduction from Section 3 can be naturally adapted to the NTRU setting. Given the components in Sections 5 to 7, we present the anonymous credentials scheme in Section 8 along with security proofs and concrete instantiations.

2 Preliminaries

Let λ be a security parameter which is provided in unary to all involved algorithms. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Let $\mathbb{Z}_p$ denote the ring of integers modulo p. We write $\vec{v} \in \mathbb{Z}_p^m$ to denote vectors over a ring $\mathbb{Z}_p$. Matrices over $\mathbb{Z}_p$ will be written as regular capital letters. By default, all vectors are column vectors.

We define $\mathcal{U}(S)$ to be the uniform distribution on the finite set S. We write $x \leftarrow D$ when x is sampled according to the distribution D. Sometimes, we abuse the notation and write $x \leftarrow S$ to denote $x \leftarrow \mathcal{U}(S)$.

Lattices. Let $B = \{\vec{b}_1, \ldots, \vec{b}_n\} \subseteq \mathbb{R}^n$ be a set of linearly independent vectors. The n-dimensional full-rank lattice generated by B is defined as follows:

$$\Lambda = \Lambda(B) := \left\{ \sum_{i=1}^{n} c_i \vec{b}_i : c_1, \ldots, c_n \in \mathbb{Z} \right\}.$$

We denote $\tilde{B} = (\vec{b}_i')_{i \in [n]}$ to be the Gram-Schmidt orthogonalization of B. Further, define the Gram-Schmidt norm of B as $\|\tilde{B}\| := \max_{i \in [n]} \|\vec{b}_i'\|$.

Cyclotomic Rings. For a power of two $d \geq 4$ and a positive integer p, let $\mathcal{K} = \mathbb{Q}[X]/(X^d + 1)$ denote the $2d$-th cyclotomic field and $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ be the corresponding ring of integers. Lower-case letters denote

elements in $\mathcal{R}$ and bold lower-case (resp. upper-case) letters represent column vectors (resp. matrices) with coefficients in $\mathcal{R}$. For a modulus $q \in \mathbb{N}$, we define $\mathcal{R}_q := \mathcal{R}/(q) = \mathbb{Z}_q[X]/(X^d+1)$. Further, we define $\mathcal{R}_q^\times$ to be the set of polynomials in $\mathcal{R}_q$ which are invertible over $\mathcal{R}_q$. For $\eta \in \mathbb{N}$, define $S_\eta := \{x \in \mathcal{R} : \|x\|_\infty \leq \eta\}$. Recall that, for any $\mathbf{A} \in \mathcal{R}_q^{n\times m}$, the q-ary lattice $\Lambda_q^\perp(\mathbf{A})$ is defined as $\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$.

Coefficient representation and multiplication matrices. For a polynomial vector $\mathbf{x} \in \mathcal{R}^l$, define $\mathsf{Coeffs}(\mathbf{x}) \in \mathbb{Z}^{ld}$ to be coefficient vector of $\mathbf{x}$. Similarly, $\mathsf{Coeffs}^{-1}(\vec{x}) \in \mathcal{R}^l$ is the polynomial vector with coefficients $\vec{x}$.

For a polynomial $f = f_0 + f_1 X + \ldots + f_{d-1}X^{d-1} \in \mathcal{R}$, we define the *multiplication* matrix $\mathsf{rot}(f) \in \mathbb{Z}^{d\times d}$ as:

$$\mathsf{rot}(f) = \begin{bmatrix} f_0 & -f_{d-1} & \ldots & -f_1 \\ f_1 & f_0 & \ldots & -f_2 \\ \vdots & \vdots & \vdots & \vdots \\ f_{d-1} & \ldots & f_1 & f_0 \end{bmatrix}.$$

In particular, we will use the property that for $f, g \in \mathcal{R}$, $\mathsf{rot}(fg) = \mathsf{rot}(f)\mathsf{rot}(g)$. We extend this definition for matrices over $\mathcal{R}$. Namely, for a matrix $\mathbf{F} = (f_{i,j}) \in \mathcal{R}_q^{n\times m}$, we define

$$\mathsf{rot}(\mathbf{F}) = \begin{bmatrix} \mathsf{rot}(f_{1,1}) & \mathsf{rot}(f_{1,2}) & \ldots & \mathsf{rot}(f_{1,m}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathsf{rot}(f_{n,1}) & \mathsf{rot}(f_{n,2}) & \ldots & \mathsf{rot}(f_{n,m}) \end{bmatrix} \in \mathbb{Z}^{nd\times md}.$$

Then, for any polynomial vector $\mathbf{x} \in \mathcal{R}_q^m$ we have the following property over $\mathbb{Z}_q$:

$$\mathsf{Coeffs}(\mathbf{F}\mathbf{x}) = \mathsf{rot}(\mathbf{F})\mathsf{Coeffs}(\mathbf{x}).$$

Discrete Gaussian distribution. We recall the discrete Gaussian distribution used for the rejection sampling and trapdoor sampling.

Definition 2.1. *The n-dimensional Gaussian function $\rho_{\mathfrak{s},\vec{c}} : \mathbb{R} \to (0,1]$ is defined by*

$$\rho_{\mathfrak{s},\vec{c}}(\vec{x}) := \exp\left(-\frac{\|\vec{x}-\vec{c}\|^2}{2\mathfrak{s}^2}\right).$$

For any coset $\Lambda + \vec{t}$ of a full-rank lattice $\Lambda \subset \mathbb{R}^n$, $\rho_{\mathfrak{s},\vec{c}}(\Lambda+\vec{t}) := \sum_{\vec{x}\in\Lambda+\vec{t}} \rho_{\mathfrak{s},\vec{c}}(\vec{x})$. Then, the discrete Gaussian distribution over a coset of a lattice $\Lambda + \vec{t}$ centred around $\vec{c} \in \mathbb{R}^n$ with standard deviation $\mathfrak{s} > 0$ is given by

$$\forall \vec{x} \in \Lambda + \vec{t}, D_{\Lambda+\vec{t},\mathfrak{s},\vec{c}}(\vec{x}) := \frac{\rho_{\mathfrak{s},\vec{c}}(\vec{x})}{\rho_{\mathfrak{s},\vec{c}}(\Lambda+\vec{t})}.$$

We write $D^n_{\mathfrak{s},\vec{c}}$ when $\Lambda = \mathbb{Z}^n$. Similarly, we ignore the subscript $\vec{c}$ when the distribution is centred around $\vec{0} \in \mathbb{Z}^n$.

Smoothing parameter. We recall the definition of a smoothing parameter [MR07]. Namely, for any n-dimensional lattice Λ and a real number $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest $s > 0$ such that $\rho_{\frac{1}{s\sqrt{2\pi}}}(\Lambda^*\backslash\{0\}) \leq \epsilon$. We also consider the scaled version $\eta'_\epsilon(\Lambda) = \frac{1}{\sqrt{2\pi}}\eta_\epsilon(\Lambda)$. Further, for any $\epsilon > 0$ we define $\eta_{\min}(\epsilon) \in \mathbb{R}$ to be such that:

$$\Pr_{\mathbf{A}\leftarrow\mathcal{R}_q^{n\times m}}[\eta'_\epsilon(\Lambda_q^\perp(\mathbf{A})) > \eta_{\min}(\epsilon)] \leq 2^{-d}.$$

Note that $\eta_{\min}$ can be computed as in [BTT22, Lemma 2.5].

Tail Bounds We recall the following tail bounds which are widely used in the literature. The first one focuses on discrete Gaussians over integers and follows from [Ban93, Lemma 1.5(i)] and was adapted in [Lyu12, Lemma 4.4]. The next one by Micciancio and Regev [MR07, Lemma 4.4] is a tail bound on discrete Gaussians over any full-rank lattices.

Lemma 2.2 ([Lyu12]). *Let $\vec{x} \leftarrow D_{\mathfrak{s}}^n$ and $t > 1$. Then*

$$\Pr\left[\|\vec{x}\| > t \cdot \mathfrak{s}\sqrt{n}\right] < \left(te^{\frac{1-t^2}{2}}\right)^n.$$

Lemma 2.3 ([MR07]). *Let Λ be an n-dimensional full-rank lattice, $\vec{c} \in \mathbb{R}^n$, $0 < \epsilon < 1$ and $\mathfrak{s} \geq \eta'_\epsilon(\Lambda)$. Let $\vec{x} \leftarrow D_{\Lambda,\mathfrak{s},\vec{c}}$. Then $\Pr\left[\|\vec{x} - \vec{c}\| \geq \mathfrak{s}\sqrt{n}\right] < \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$.*

Preimage sampling. Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$. Then, we denote $\mathbf{A}_{\mathfrak{s}}^{-1}(\mathbf{u})$ to be the random variable $\mathbf{x} \leftarrow D_{\mathfrak{s}}^{md}$ conditioned on $\mathbf{Ax} = \mathbf{u}$ over $\mathcal{R}_q$.

Rejection sampling. Rejection sampling [Lyu09; Lyu12] is a widely used technique to ensure the zero-knowledge property of many lattice-based (non-)interactive proofs.

Lemma 2.4 (Rejection Sampling [Lyu12]). *Let $V \subseteq \mathcal{R}^\ell$ be a set of polynomials with norm at most T and $\rho\colon V \to [0,1]$ be a probability distribution. Fix the standard deviation $\mathfrak{s} = \alpha T$ for $\alpha = O(\sqrt{\lambda})$. Let*

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

Now, sample $\mathbf{v} \leftarrow \rho$ and $\mathbf{y} \leftarrow D_{\mathfrak{s}}^\ell$, set $\mathbf{z} = \mathbf{y} + \mathbf{v}$, and run $b \leftarrow \mathsf{Rej}_1(\mathbf{z}, \mathbf{v}, \mathfrak{s}, M)$ as defined in Figure 1. Then, the probability that $b = 1$ is at least $(1 - 2^{-\lambda})/M$ and the distribution of $(\mathbf{v}, \mathbf{z})$, conditioned on $b = 1$, is within statistical distance of $2^{-\lambda}$ of the product distribution $\rho \times D_{\mathfrak{s}}^\ell$.

$\mathsf{Rej}\,(\vec{z}, \vec{v}, \mathfrak{s}, M)$

1: $u \leftarrow [0,1)$
2: **if** $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$ **then**
3: **return** 0 (i.e. *reject*)
4: **else**
5: **return** 1 (i.e. *accept*)

Fig. 1: Standard rejection sampling algorithm [Lyu12].

Recently, Boschini et al. [BTT22] proposed a generalized rejection sampling for elliposidal Gaussians over any lattice which was later used in the context of multi-signatures. Here, we use the (simplified) result from [BTT22, Theorem B.1] to apply rejection sampling on q-ary lattices[9].

Lemma 2.5 (Generalized Rejection Sampling [BTT22]). *Take any $\alpha, T > 0$ and $\epsilon \leq 1/2$. Let $\mathbf{v} \in \mathcal{R}_q^m$ be such that $\|\mathbf{v}\| \leq T$, $\mathbf{A} \in \mathcal{R}_q^{n \times m}, \mathbf{w} \in \mathcal{R}_q^n$ and $\mathbf{t} := \mathbf{Av} \in \mathcal{R}_q^n$. Also, pick $\mathfrak{s} \geq \max(\alpha T, \eta'_\epsilon(\Lambda_q^\perp(\mathbf{A})))$. Then, for any*

$$t > 0, \quad M := \exp\left(\frac{1}{2\alpha^2} + \frac{t}{\alpha}\right), \quad \varepsilon := 2\left(\frac{1+\epsilon}{1-\epsilon}\right)\exp\left(-2t^2 \cdot \frac{\pi - 1}{\pi}\right),$$

the statistical distance between distributions $\mathsf{RejSamp}$ and SimRS defined in Figure 2 is at most $\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}$. Moreover, the probability that $\mathsf{RejSamp}$ outputs something is at least $\frac{1-\varepsilon}{M}\left(1 - \frac{4\epsilon}{(1+\epsilon)^2}\right)$.

[9] An almost identical application was described in [BTT22, Section B.4] in the context of proving statistical honest-verifier zero-knowledge of the Fiat-Shamir with aborts protocol [Lyu12].

$\underline{\mathsf{RejSamp}(\mathbf{A}, \mathbf{v}, \mathbf{t}, \mathbf{w})}$

1: **if** $\mathbf{Av} \neq \mathbf{t}$ **then return** $\perp$
2: $\mathbf{y} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}(\mathbf{w})$
3: $\mathbf{z} = \mathbf{y} + \mathbf{v}$
4: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, \mathbf{z})$ with prob. $\min\left(\frac{D_{\mathfrak{s}}^{md}(\mathbf{z})}{M \cdot D_{\mathfrak{s},\mathbf{v}}^{md}(\mathbf{z})}, 1\right)$

$\underline{\mathsf{SimRS}(\mathbf{A}, \mathbf{t}, \mathbf{w})}$

1: **if** $\mathbf{Av} \neq \mathbf{t}$ **then return** $\perp$
2: $\mathbf{z} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}(\mathbf{t} + \mathbf{w})$
3: **return** $(\mathbf{A}, \mathbf{t}, \mathbf{w}, \mathbf{z})$ with prob. $\frac{1}{M}$

Fig. 2: Rejection sampling on q-ary lattices.

2.1 NTRU Lattices

Using terminology from above, let d be a power of two, q a positive integer and $f, g \in \mathcal{R}$ such that f is invertible over $\mathcal{R}_q$. Let $h = g/f \in \mathcal{R}_q$. The NTRU lattice associated to h and q is defined as

$$\Lambda_{h,q} := \{(u, v) \in \mathcal{R}^2 : u + vh = 0 \bmod q\} \ .$$

Then, $\Lambda_{h,q}$ is a $2d$-dimensional full-rank lattice generated by the rows of

$$A_{h,q} := \begin{bmatrix} -\mathsf{rot}(h) & I_d \\ q \cdot I_d & 0 \end{bmatrix} \in \mathbb{Z}^{2d \times 2d}.$$

We recall that there is an efficient algorithm $\texttt{NTRU.TrapGen}$, which given modulus q and the ring dimension d, outputs $h \in \mathcal{R}_q$ and a short basis of $\Lambda_{h,q}$. This is the core part of the key generation of the Falcon signature scheme [PFH+17].

Lemma 2.6 ([DLP14; PFH+17]). *There is a PPT algorithm* $\texttt{NTRU.TrapGen}(q, d)$ *which outputs* $h \in \mathcal{R}_q$ *and a basis* $\mathbf{B}$ *of* $\Lambda_{h,q}$ *such that* $\|\tilde{\mathbf{B}}\| \leq 1.17\sqrt{q}$.

The short basis can now be used for preimage sampling using the well-known GPV framework [GPV08] and its concrete instantiation in [DLP14]. Namely, for any $c \in \mathcal{R}$, one can efficiently sample $(u, v) \in \mathcal{R}^2$ from a discrete Gaussian distribution conditioned on $u + vh = c \bmod q$. We use the extended result following [WW22, Lemma 2.7], i.e. given additionally $\mathbf{a} \in \mathcal{R}^m$, we can efficiently sample $(u, v, \mathbf{w}) \in \mathcal{R}^{m+2}$ from a discrete Gaussian distribution conditioned on $u + vh + \mathbf{a}^T\mathbf{m} = c \bmod q$.

Lemma 2.7 ([DN12; DLP14; GPV08]). *Let* $n \in \mathbb{N}$ *and* $\epsilon = 2^{-\lambda}/(4d)$. *There is a PPT algorithm* $\texttt{GSampler}$, *which takes* $(h, \mathbf{B}) \leftarrow \texttt{NTRU.TrapGen}(q, d)$, $\mathbf{a} \in \mathcal{R}_q^m$, *standard deviation* $\mathfrak{s} > 0$ *and a target vector* $c \in \mathcal{R}_q$ *as input, and outputs a triple* $(u, v, \mathbf{w}) \in \mathcal{R}_q^{m+2}$ *such that*

$$\Delta\left(\left[h\ \mathbf{a}^T\ 1\right]_{\mathfrak{s}}^{-1}(c), \texttt{GSampler}(h, \mathbf{a}, \mathbf{B}, \mathfrak{s}, c)\right) \leq 2^{-\lambda}$$

as long as

$$\mathfrak{s} \geq 1.17\sqrt{q} \cdot \eta'_\epsilon(\mathbb{Z}) \quad \textit{where} \quad \eta'_\epsilon(\mathbb{Z}) \approx \frac{1}{\pi} \cdot \sqrt{\frac{1}{2}\ln\left(2 + \frac{2}{\epsilon}\right)}.$$

2.2 Module-SIS and Module-LWE Problems

The security of our schemes relies on the well-known computational lattice problems Module-LWE (MLWE) and Module-SIS (MSIS) [LS15]. Both problems are defined over $\mathcal{R}_q$.

Definition 2.8 ($\mathsf{MSIS}_{n,m,\mathcal{B}}$). *Given* $A \leftarrow \mathcal{R}_q^{n \times m}$, *the* Module-SIS *problem with parameters* $n, m > 0$ *and* $0 < \mathcal{B} < q$ *asks to find* $\mathbf{z} \in \mathcal{R}_q^m$ *such that* $A\mathbf{z} = \mathbf{0}$ *over* $\mathcal{R}_q$ *and* $0 < \|\mathbf{z}\| \leq \mathcal{B}$. *An algorithm* $\mathcal{A}$ *is said to have advantage* ϵ *in solving* $\mathsf{MSIS}_{n,m,\mathcal{B},q}$ *if*

$$\mathsf{Adv}^{\mathsf{MSIS}}_{n,m,\mathcal{B},q}(\mathcal{A}) := \Pr\left[0 < \|\mathbf{z}\|_\infty \leq \mathcal{B} \wedge A\mathbf{z} = \mathbf{0} \,\middle|\, A \leftarrow \mathcal{R}_q^{n \times m};\ \mathbf{z} \leftarrow \mathcal{A}(A)\right] \geq \epsilon.$$

If the modulus is clear from the context, then we drop q *from the subscript.*

As for Module-LWE, we consider its "knapsack version" which is equivalent to its standard definition up to an additive negligible factor for typically chosen parameters [EZS+19, Appendix C]. In the paper we will use both versions and assume they are equivalent.

Definition 2.9 ($\mathsf{MLWE}_{n,m,\chi,q}$). *The* Module-LWE *problem with parameters $m \geq n > 0$ and an error distribution χ over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases: 1) $(A, A\boldsymbol{s})$ for $A \leftarrow \mathcal{R}_q^{n\times m}$, a secret vector $\boldsymbol{s} \leftarrow \chi^m$, and 2) $(A, \boldsymbol{b}) \leftarrow \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^n$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MLWE}_{n,m,\chi,q}$ if*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MLWE}}_{n,m,\chi,q}(\mathcal{A}) := \Big|\Pr\Big[b = 1 \,\Big|\, A \leftarrow \mathcal{R}_q^{n\times m};\ \boldsymbol{s} \leftarrow \chi^m;\ b \leftarrow \mathcal{A}(A, A\boldsymbol{s})\Big] \\ - \Pr\Big[b = 1 \,\Big|\, A \leftarrow \mathcal{R}_q^{n\times m};\ \boldsymbol{b} \leftarrow \mathcal{R}_q^n;\ b \leftarrow \mathcal{A}(A, \boldsymbol{b})\Big]\Big| \geq \epsilon.\end{aligned} \tag{5}$$

For simplicity, if χ is a uniform distribution over S_η then we simply write $\mathsf{MLWE}_{n,m,\eta,q}$. Also, we drop the subscript q if the modulus is clear from the context.

2.3 Non-Interactive Zero-Knowledge Proofs in the ROM

In this paper, we consider binary relations $R \subseteq \{0,1\}^* \times \{0,1\}^*$. Then define $L_R := \{\mathsf{x} \in \{0,1\}^* : \exists \mathsf{w}, (\mathsf{x}, \mathsf{w}) \in R\}$ be the language corresponding to R. We refer to x as a statement and w as a witness.

We recall the (slightly adapted) definitions of non-interactive zero-knowledge proofs (NIZK) in the random oracle model from [PK22].

Definition 2.10 (NIZK). *A non-interactive zero-knowledge proof system Π_{NIZK} for the relation R consists of three PPT algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ which are defined as follows:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$ *: the setup algorithm which outputs the common reference string* $\mathsf{crs} \in \{0,1\}^{\ell(\lambda)}$,
- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w) \to \pi/\perp$*: the prover algorithm takes as input the common reference string* $\mathsf{crs} \in \{0,1\}^\ell$*, statement x and witness w, either outputs a proof π or an abort symbol $\perp$,*
- $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi) \to 0/1$*: the verifier algorithm takes as input the common reference string* $\mathsf{crs} \in \{0,1\}^\ell$*, statement x and a proof π and outputs a bit b where $b = 1$ means accept and $b = 0$ means reject.*

Unless stated otherwise, we assume that crs can be generated as the output of another random oracle. In other words, crs is a common random string. Hence, our protocols do not require a trusted setup.

We recall the key properties of NIZK used in this paper: (i) correctness, (ii) zero-knowledge, and (iii) multi-proof extractability (i.e. straight-line extractability).

Definition 2.11 (Correctness). *We say that a NIZK proof system Π_{NIZK} is correct if for all $\mathsf{crs} \in \{0,1\}^\ell$ and $(\mathsf{crs}, x, w) \in R$, the probability that $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ outputs $\perp$ is $\mathsf{negl}(\lambda)$, and:*

$$\Pr\left[\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w) : \mathsf{Verify}(\mathsf{crs}, x, w) = 1 \middle| \pi \neq \perp\right] = 1 - \mathsf{negl}(\lambda).$$

Definition 2.12 (Zero-Knowledge). *We say that a NIZK proof system Π_{NIZK} is zero-knowledge if there exists a simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ which consists of two PPT algorithms with a shared state such that for any PPT adversary $\mathcal{A}$ we have*

$$\mathsf{Adv}^{\mathsf{ZK}}_{\Pi_{\mathsf{NIZK}}}(\mathcal{A}) := \Big|\Pr[1 \leftarrow \mathcal{A}^{\mathsf{H},\mathsf{Prove}}(\mathsf{crs})] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{S}_0,\mathcal{S}_1}(\mathsf{crs})]\Big| = \mathsf{negl}(\lambda)$$

where Prove and $\mathcal{S}$ are prover and simulator oracles which, given (x, w), output $\perp$ if $(\mathsf{crs}, x, w) \notin R$ and otherwise return $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ and $\mathcal{S}_1(\mathsf{crs}, x)$ respectively. The probability is also taken over the randomness of generating the common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$.

Finally, we consider multi-proof extractability which corresponds to straight-line extractability. Here, the adversary can pick the statements adaptively. In order to perform extraction in this stronger setting, the common reference string is simulated and the corresponding trapdoor is provided to the extractor.

Definition 2.13 (Multi-Proof Extractability). *The following hold:*

CRS Simulatability. For any PPT adversary $\mathcal{A}$, we have:

$$\mathsf{Adv}^{\mathsf{crs}}_{\Pi_{\mathsf{NIZK}}}(\mathcal{A}) := \left|\Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) : 1 \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{crs})] - \Pr[(\tilde{\mathsf{crs}}, \tau) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda) : 1 \leftarrow \mathcal{A}^{\mathsf{H}}(\tilde{\mathsf{crs}})]\right|$$

is negligible.

Straight-Line Extractability. There exist constants e_1, e_2, c such that for any $Q_{\mathsf{H}}, Q_s \in \mathsf{poly}(\lambda)$ and any PPT adversary $\mathcal{A}$ that makes at most Q_{H} random oracle queries with

$$\Pr\left[\begin{array}{c}(\tilde{\mathsf{crs}}, \tau) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(x_i, \pi_i)\}_{i\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\tilde{\mathsf{crs}})\end{array} : \forall i \in [Q_s], \mathsf{Verify}^{\mathsf{H}}(\tilde{\mathsf{crs}}, x_i, \pi_i) = 1\right] \geq \varepsilon(\lambda),$$

where $\varepsilon(\lambda)$ is non-negligible, we have

$$\Pr\left[\begin{array}{c}(\tilde{\mathsf{crs}}, \tau) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \{(x_i, \pi_i)\}_{i\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\tilde{\mathsf{crs}}), \\ \{w_i \leftarrow \mathit{Multi\text{-}Extract}(Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}}, \tau, x_i, \pi_i)\}_{i\in[Q_s]} : \\ \forall i \in [Q_s], (\tilde{\mathsf{crs}}, x_i, w_i) \in R \wedge \mathsf{Verify}^{\mathsf{H}}(\tilde{\mathsf{crs}}, x_i, \pi_i)\end{array}\right] \geq \frac{1}{2}\cdot\varepsilon(\lambda) - \mathsf{negl}(\lambda)$$

where the runtime of the extractor is upper-bounded by $Q_{\mathsf{H}}^{e_1} \cdot Q_s^{e_2} \cdot \frac{1}{\varepsilon(\lambda)^c} \cdot \mathsf{poly}(\lambda)$.

For proving various soundness properties, we will deal with *expected* PPT knowledge extractors. Since security of our protocols rely on computation assumptions, it is essential for us to transform these extractors into *strict* PPT algorithms. Below, we show a standard way to achieve this goal.

Lemma 2.14 (Expected-Time to Strict-Time Transformation). *Take any binary relation R and any statement x. Let $\mathcal{A}$ be a probabilistic algorithm which runs in expected time at most T and*

$$\Pr_{\mathsf{crs}\leftarrow\{0,1\}^\ell}[\mathcal{A}(\mathsf{crs}, x) \to w : (x, w) \in R] = \varepsilon.$$

Then, there is an algorithm $\mathcal{B}$ with an oracle access to $\mathcal{A}$, which runs in time at most $\frac{2T}{\varepsilon}$ and

$$\Pr_{\mathsf{crs}\leftarrow\{0,1\}^\ell}[\mathcal{B}^{\mathcal{A}}(\mathsf{crs}, x) \to 1] \geq \varepsilon/2.$$

Proof. The algorithm $\mathcal{B}$ is pretty intuitive: given input (crs, x), it runs $\mathcal{A}(\mathsf{crs}, x)$ and checks whether $\mathcal{A}$ did output a valid witness w. However, if the total runtime is more than $T_{\max} := 2T/\varepsilon$, then $\mathcal{B}$ aborts.

Denote X to be the binary random variable, which determines whether the output w from $\mathcal{A}$ is valid. Similarly, let Y be the runtime of $\mathcal{A}(\mathsf{crs}, x)$. Then, we are interested in

$$\Pr[X = 1 \wedge Y \leq T_{\max}] \geq \Pr[X = 1] - \Pr[Y > T_{\max}].$$

Note that $\Pr[X = 1] = \varepsilon$, and also by Markov inequality we have $\Pr[Y > T_{\max}] \leq \varepsilon/2$. Hence, this concludes the proof. □

3 The ISIS_f Assumption

This section introduces a new family of lattice-based assumptions called ISIS_f (where ISIS stands for Inhomogenous Shortest Integer Solution). In a similar flavour as the recently proposed lattice assumptions [ACL+22; ASY21], the adversary is given short preimages of random (but not necessarily uniformly random) images and its task is to come up with either a short preimage of a new image, or a new preimage (i.e. not the one received earlier) to one of the images which was sent. More formally, let $\mathsf{n}, \mathsf{m}, k \in \mathbb{N}$ where the first two variables correspond to the matrix dimensions and the last one represents the number of samples. We consider a uniformly random $\mathsf{n} \times \mathsf{m}$ matrix $\mathbf{A}$ over $\mathcal{R}_q$, and an efficiently computable function $f : [N] \to \mathbb{Z}_q^n$. The adversary is given $x_1, \ldots, x_k \leftarrow [N]$ as well as vectors $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \mathbb{Z}_q^m$, where each $\mathbf{s}_i \leftarrow \mathbf{A}_{\mathbf{s}}^{-1}(f(x_i))$. Then, it needs to come up with a new pair $(x^*, \mathbf{s}^*)$ (in particular, different from the previous ones) such that $\mathbf{A}\mathbf{s}^* = f(x^*)$ and $\mathbf{s}^*$ is short.

$\underline{\mathsf{Exp}^{\mathsf{ISIS}_f}_{\mathsf{pp}}(\mathcal{A})}$

$\mathbf{A} \leftarrow \mathcal{R}_q^{\mathsf{n}\times\mathsf{m}}$
for $i \in [k]$:
  $x_i \leftarrow [N]$
  $\mathbf{s}_i \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}(f(x_i))$
$(x^*, \mathbf{s}^*) \leftarrow \mathcal{A}(\mathbf{A}, (x_i, \mathbf{s}_i)_{i\in[k]})$
if $\big(\mathbf{A}\mathbf{s}^* = f(x^*)\big) \wedge \big(0 < \|\mathbf{s}^*\| \leq \mathcal{B}\big) \wedge \big((x^*, \mathbf{s}^*) \notin \{(x_i, \mathbf{s}_i) : i \in [k]\}\big)$
  then return 1
else return 0

Fig. 3: The ISIS_f experiment.

Definition 3.1 (The ISIS_f Problem). *Let $\mathsf{pp} := (q, d, \mathsf{n}, \mathsf{m}, N, k, \mathfrak{s}, \mathcal{B})$ be a tuple of functions of the security parameter λ. Consider any efficiently computable function $f : [N] \to \mathcal{R}_q^{\mathsf{n}}$. The ISIS_f assumption is defined by the experiment in Figure 3. For an adversary $\mathcal{A}$, we define*

$$\mathsf{Adv}^{\mathsf{ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) = \Pr[\mathsf{Exp}^{\mathsf{ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) \to 1].$$

The $\mathsf{ISIS}_f^{\mathsf{pp}}$ assumption states that for every PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{ISIS}_f}_{\mathsf{pp}}(\mathcal{A})$ is negligible.

3.1 Concrete Instantiations of f

3.1.1 Random Function

An intuitive choice for $f = \mathsf{RF}$ is a random function, modelled in the security analysis as a random oracle. In this case, if $N = \omega(\mathsf{poly}(\lambda))$ then following the GPV signature proof [GPV08, Proposition 6.1], ISIS_f can be tightly reduced to the plain Module-SIS assumption. Unfortunately, the issue with picking f to be random is that in our constructions we need to prove knowledge of a preimage of f. This becomes slightly awkward since then we would require proof systems that can actually prove statement related to the random oracle f.

3.1.2 Binary Encoding

In our constructions we will use the following family of functions f. Let $t \in \mathbb{N}$ and take any matrix $B \in \mathbb{Z}_q^{nd\times\mathsf{t}}$. Then, the function is defined as:

$$f(x) := \mathsf{Coeffs}^{-1}(B \cdot \mathsf{enc}(x)) \in \mathcal{R}_q^{\mathsf{n}}$$

where $\mathsf{enc}(x) \in \{0,1\}^t$ is a binary decomposition of $x-1$. Hence, naturally $N = 2^{\mathsf{t}}$. Clearly, proving knowledge of a preimage x of f, i.e. $f(x) = \mathbf{y}$, is equivalent to proving knowledge of a binary vector $\vec{u} \in \{0,1\}^{\mathsf{t}}$ such that $B\vec{u} = \mathsf{Coeffs}(\mathbf{y})$. This, in turn, can be proven using the [LNP22] framework. The main purpose of the matrix B is to provide flexibility when setting N and has no significant impact on the security (apart from a few naive special cases). We denote the corresponding ISIS_f problem as $\mathsf{ISIS}_{\mathsf{bin}}$. In the following, we propose a few standard attacks on $\mathsf{ISIS}_{\mathsf{bin}}$.

Relations to finding a short vector in the lattice. Regardless of the choice of f, one of the most naive attacks would be to simply try any x and then use the lattice attacks to find a short vector $\mathbf{s}^*$ such that

$$\mathbf{A}\mathbf{s}^* = f(x)$$

which has a flavour of the Inhomogenous-MSIS problem. Now, if B is a zero (resp. identity) matrix then this corresponds to the plain (resp. Hermite Normal Form) Module-SIS [DKL+18].

We propose another attack, which this time actually makes use of the k pairs $(x_i, \mathbf{s}_i)_{i\in[k]}$. Denote $\vec{u}_i = \mathsf{enc}(x_i)$ and $\vec{s}_i = \mathsf{Coeffs}(\mathbf{s}_i)$ for $i \in [k]$. Then, by definition of $\mathsf{ISIS}_{\mathsf{bin}}$ we have [10]:

$$\left[\mathsf{rot}(\mathbf{A}) \; -B\right] S = 0 \quad \text{where} \quad S := \begin{bmatrix} \vec{s}_1 & \cdots & \vec{s}_k \\ \vec{u}_1 & \cdots & \vec{u}_k \end{bmatrix} \in \mathbb{Z}_q^{(\mathsf{m}d+\mathsf{t})\times k}. \tag{6}$$

Here, the attack is to consider the lattice Λ generated by columns of S and find a short vector $(\vec{s}^*, \vec{u}^*) \in \Lambda$, such that $\|\vec{s}^*\| \leq \mathcal{B}, \vec{u}^* \in \{0,1\}^{\mathsf{t}}$ and $(\vec{s}^*, \vec{u}^*) \notin S$. We briefly explain why this allows us to solve $\mathsf{ISIS}_{\mathsf{bin}}$. Suppose we found a vector $(v_1, \ldots, v_k) \in \mathbb{Z}_q^k$ such that

$$\vec{s}^* = \sum_{i=1}^{k} v_i \vec{s}_i \quad \text{and} \quad \vec{u}^* = \sum_{i=1}^{k} v_i \vec{u}_i$$

where $\vec{s}^*$ and $\vec{u}^*$ satisfy all the conditions above. Then, for $\mathbf{s}^* = \mathsf{Coeffs}^{-1}(\vec{s}^*)$ we have

$$\mathbf{A}\mathbf{s}^* = \sum_{i=1}^{k} v_i \mathbf{s}^* = \sum_{i=1}^{k} v_i \mathsf{Coeffs}^{-1}(B\vec{u}_i) = \mathsf{Coeffs}^{-1}\left(\sum_{i=1}^{k} v_i \cdot B\vec{u}_i\right) = \mathsf{Coeffs}^{-1}(B\vec{u}^*).$$

Finally, by the conditions above, this gives us a valid $\mathsf{ISIS}_{\mathsf{bin}}$ solution.

A naive solution to obtain $(\vec{s}^*, \vec{u}^*)$ would be to hope that S has full-rank $\mathsf{m}d + \mathsf{t}$ for large enough instances k. Then, the adversary could pick any valid $(\vec{s}^*, \vec{u}^*)$ and compute the linear combination $v_1, \ldots, v_k \in \mathbb{Z}_q$ using linear algebra. The hardness of this problem lies in the fact that by (6) and the rank-nullity theorem, we have that the rank of S is at most $\mathsf{n}d$. Since in all our applications $\mathsf{n} \leq \mathsf{m}$, this implies that S will never have rank at least $\mathsf{m}d + \mathsf{t}$.

Relations to integer linear programming. Let us denote $\vec{u}_i = \mathsf{enc}(x_i)$ for $i \in [k]$. The attack involves finding a short non-zero vector $\vec{v} \in \mathbb{Z}_q^k$ such that

$$\sum_{i=1}^{k} v_i \vec{u}_i = \vec{0} \pmod{q}.$$

Note that if $\vec{u}_1, \ldots, \vec{u}_k$ as well as $\vec{v}$ are sufficiently short w.r.t. q then the equation above holds over integers. Now, we can use techniques from integer linear programming (ILP) [BDE+18; HM17] to find such a vector $\vec{v}$. Especially, Herold and May [HM17, Theorem 1] showed that under some mild assumptions, one can efficiently find such a binary vector $\vec{v} \in \{0,1\}^k$, as long as $k \geq 2\mathsf{t}$. Let $i \in [k]$ such that $v_i \neq 0$. Then, if we denote $\mathbf{s}^* := -(\sum_{j\neq i} v_j \mathbf{s}_j)$, then

$$\mathbf{A}\mathbf{s}^* = -\sum_{j\neq i} v_j \mathbf{A}\mathbf{s}_j = -\sum_{j\neq i} v_j \mathsf{Coeffs}^{-1}(B\vec{u}_j) = v_i \mathsf{Coeffs}^{-1}(B\vec{u}_i) = \mathsf{Coeffs}^{-1}(B\vec{u}_i).$$

If $\mathfrak{s}$ is larger than the norm of the shortest norm of $\Lambda_q^{\perp}(\mathbf{A})$, then by the unpredictability argument (c.f. Lemma 3.8), with non-negligible probability we have $\mathbf{s}^* \neq \mathbf{s}_i$. Now, using the standard tail bounds (c.f. Section 2), we can upper-bound the norm of $\mathbf{s}^*$ by $(k-1) \cdot \mathfrak{s}\sqrt{\mathsf{m}d}$. Hence, if $\mathcal{B}$ is larger than that value then $(\mathbf{s}^*, f^{-1}(\mathbf{b}_i))$ is a valid $\mathsf{ISIS}_{\mathsf{bin}}$ solution[11]. Therefore, in order to prevent such attacks, we need to set $\mathcal{B} \approx \mathfrak{s}\sqrt{\mathsf{m}d}$ to be close to the tail bound of the discrete Gaussian distribution with standard deviation $\mathfrak{s}$.

[10] We recall the skew-circulant matrices and the function rot in Section 2.

[11] Obviously, one could try to use ILP for smaller k to decrease the bound on $\mathbf{s}^*$.

$\underline{\mathsf{Exp}^{\mathsf{Int\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A})}$

$\mathbf{A} \leftarrow \mathcal{R}_q^{\mathsf{n}\times\mathsf{m}}$
$\mathbf{C} \leftarrow \mathcal{R}_q^{\mathsf{n}\times(\ell_m+\ell_r)}$
$\mathcal{M} = \emptyset$
$(x^*, \mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{pre}}}(\mathbf{A}, \mathbf{C})$
if $(\mathbf{m}^* \notin \mathcal{M}) \wedge \left(\mathbf{A}\mathbf{s}^* = f(x^*) + \mathbf{C}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}\right) \wedge \left(0 < \|\mathbf{s}^*\| \le \mathcal{B}_s\right) \wedge \left(\|(\mathbf{m}^*, \mathbf{r}^*)\| \le \mathcal{B}_m\right)$
  then return 1
else return 0

$\underline{\mathcal{O}_{\mathsf{pre}}(\mathbf{m}, \mathbf{r})}$

if $\|(\mathbf{m}, \mathbf{r})\| < \mathcal{B}_m$
  then return $\perp$
$x \leftarrow [N]$
$\mathbf{s} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}\left(f(x) + \mathbf{C}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}\right)$
$\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
return $(x, \mathbf{s})$

Fig. 4: Interactive version of the ISIS_f problem.

3.2 Interactive Version

We are ready to introduce the interactive version of the ISIS_f problem which will be used as an underlying cryptographic assumption of our anonymous credentials. The main difference is that we allow the adversary to make *preimage queries* with respect to adaptively chosen messages, i.e. it has access to the preimage oracle $\mathcal{O}_{\mathsf{pre}}$. Here, $\mathcal{O}_{\mathsf{pre}}$ on input $(\mathbf{m}_i, \mathbf{r}_i)$ first samples $x_i \leftarrow [N]$ and then returns x_i along with

$$\mathbf{s}_i \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}\left(f(x_i) + \mathbf{C}\begin{bmatrix}\mathbf{m}_i\\ \mathbf{r}_i\end{bmatrix}\right)$$

where $\mathbf{C}$ is a uniformly random public matrix. The adversary's goal is to come up with a new tuple $(x^*, \mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*)$ such that $\mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*$ are short vectors,

$$\mathbf{A}\mathbf{s}^* = f(x^*) + \mathbf{C}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix} \tag{7}$$

and $\mathbf{m}^* \notin \{\mathbf{m}_1, \ldots, \mathbf{m}_k\}$ where k is the number of oracle queries. In order to avoid trivial attacks, we require the inputs $(\mathbf{m}_i, \mathbf{r}_i)$ to the preimage oracle $\mathcal{O}_{\mathsf{pre}}$ to be short vectors as well.

Definition 3.2 (The Interactive ISIS_f Problem). *Define public parameters* $\mathsf{pp} := (q, d, \mathsf{n}, \mathsf{m}, \ell_m, \ell_r, N, \mathfrak{s}, \mathcal{B}_s, \mathcal{B}_m)$ *as a tuple of functions of the security parameter* λ*. Consider any efficiently computable function* $f : [N] \to \mathcal{R}_q^{\mathsf{n}}$*. The* $\mathsf{Int\text{-}ISIS}_f$ *assumption is defined by the experiment in Figure 4. For an adversary* $\mathcal{A}$*, we define*

$$\mathsf{Adv}^{\mathsf{Int\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) = \Pr[\mathsf{Exp}^{\mathsf{Int\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) \to 1].$$

The $\mathsf{Int\text{-}ISIS}^{\mathsf{pp}}_f$ *assumption states that for every PPT adversary* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathsf{Int\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A})$ *is negligible.*

The main result of this section is an efficient reduction from the interactive ISIS_f to the one in Definition 3.2.

Theorem 3.3 ($\mathsf{Int\text{-}ISIS}_f \implies \mathsf{ISIS}_f$). *Let* $\mathsf{pp} := (q, d, \mathsf{n}, \mathsf{m}, \ell_m, \ell_r, N, \mathfrak{s}, \mathcal{B}_s, \mathcal{B}_m)$ *be public parameters such that* $q/2 > \mathcal{B}_m \ge 1, \mathsf{m} = \mathsf{n}\log q + r$*, and*

$$M := \exp\left(1 + \frac{1}{2\alpha^2}\right) \quad \textit{and} \quad \varepsilon := 2\left(\frac{1+\epsilon}{1-\epsilon}\right)\exp\left(-2\alpha^2 \cdot \frac{\pi - 1}{\pi}\right)$$

where $\epsilon \le 1/2, \alpha \ge 1$ *and* $r > 0$*. Suppose* $\mathfrak{s} \ge \max\left(\eta_{\min}(\epsilon), \alpha\mathcal{B}_m d\sqrt{(\ell_m + \ell_r)\mathsf{m}}\right)$*. Then, for every adversary* $\mathcal{A}$ *which makes at most* Q *queries to* $\mathcal{O}_{\mathsf{pre}}$*, there is an adversary* $\mathcal{A}'$ *which runs in time essentially identical to* $\mathcal{A}$ *and*

$$\mathsf{Adv}^{\mathsf{ISIS}_f}_{\mathsf{pp}'}(\mathcal{A}') \ge \frac{1}{6Q}\mathsf{Adv}^{\mathsf{Int\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) - \frac{\ell_m + \ell_r}{12Q}\sqrt{\left(1 + 2^{-r}\right)^d - 1} - \frac{2^{-\lambda}}{6} - \frac{T_{\max}^2 Q}{12N}$$

$\underline{\mathcal{O}_1(\mathbf{m},\mathbf{r})}$
1: **if** $\|(\mathbf{m},\mathbf{r})\| > \mathcal{B}_m$
2: **then return** $\perp$
3: $\mathsf{ctr} = 0$
4: **while** $\mathsf{ctr} \geq 0$ **do**
5: $x \leftarrow [N]$
6: $\mathbf{s} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}\left(f(x) + \mathbf{C}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}\right)$
7: $u \leftarrow [0,1)$
8: **if** $u \leq \frac{1}{M}$:
9: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
10: **return** $(x,\mathbf{s})$
11: $\mathsf{ctr} = \mathsf{ctr} + 1$
12: **return** $\perp$

$\underline{\mathcal{O}_2(\mathbf{m},\mathbf{r})}$
1: **if** $\|(\mathbf{m},\mathbf{r})\| > \mathcal{B}_m$
2: **then return** $\perp$
3: $\mathsf{ctr} = 0$
4: $\rho = \perp$
5: **while** $\mathsf{ctr} < T_{\max}$ **do**
6: $x \leftarrow [N]$
7: $\mathbf{s} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}\left(f(x) + \mathbf{C}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}\right)$
8: $u \leftarrow [0,1)$
9: **if** $u \leq \frac{1}{M} \wedge \rho = \perp$:
10: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
11: $\rho = (x,\mathbf{s})$
12: $\mathsf{ctr} = \mathsf{ctr} + 1$
13: **return** ρ

$\underline{\mathcal{O}_3(\mathbf{m},\mathbf{r})}$
1: **if** $\|(\mathbf{m},\mathbf{r})\| > \mathcal{B}_m$
2: **then return** $\perp$
3: $\mathbf{v} = \mathbf{D}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}$
4: $\mathsf{ctr} = 0$
5: $\rho = \perp$
6: **while** $\mathsf{ctr} < T_{\max}$ **do**
7: $x \leftarrow [N]$
8: $\bar{\mathbf{s}} \leftarrow \mathbf{A}_{\mathfrak{s}}^{-1}(f(x))$
9: $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{v}$
10: $u \leftarrow [0,1)$
11: **if** $u \leq \min\left(\frac{D_{\mathfrak{s}}^{md}(\mathbf{s})}{M \cdot D_{\mathfrak{s},\mathbf{v}}^{md}(\mathbf{s})}, 1\right) \wedge \rho = \perp$:
12: $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
13: $\rho = (x,\mathbf{s})$
14: $\mathsf{ctr} = \mathsf{ctr} + 1$
15: **return** ρ

Fig. 5: Preimage oracles $\mathcal{O}_{\mathsf{pre}}$ used in the hybrid argument.

$$-\frac{\ell_m}{3}\cdot 2^{-rd} - \left(Q - \frac{2}{3}\right) T_{\max}\left(\frac{\varepsilon}{2M} - \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}$$

where $\mathsf{pp}' := (q, d, \mathsf{n}, \mathsf{m}, N, T_{\max}Q, \mathfrak{s}, \mathcal{B} = \mathcal{B}_s + \mathcal{B}_m d\sqrt{(\ell_m + \ell_r)\mathsf{m}})$ *and* $T_{\max}$ *satisfies* $\left(1 - \frac{1}{M}\right)^{T_{\max}} \leq 2^{-\lambda}$.

Proof. Assume without loss of generality that $\mathcal{A}$ makes exactly Q queries. We will prove the statement using a hybrid argument. Namely, in each game we will either modify the execution of the $\mathcal{O}_{\mathsf{pre}}$ oracle, or change the execution of the challenger. In the following, define E_i to be the event that $\mathcal{A}$ wins the $\mathsf{Int\text{-}ISIS}_f$ experiment.

Game_1: This is the standard Interactive ISIS_f game. Hence, $\Pr[\mathsf{E}_1] = \varepsilon$. Let us denote the preimage oracle $\mathcal{O}_0 := \mathcal{O}_{\mathsf{pre}}$.

Game_2: Here, instead of sampling $\mathbf{C}$ uniformly random, the challenger first picks a uniformly random matrix of binary polynomials $\mathbf{D} \in \mathcal{R}_q^{\mathsf{m}\times(\ell_m+\ell_r)}$ and sets $\mathbf{C} = \mathbf{AD}$.

Lemma 3.4. $\Pr[\mathsf{E}_2] \geq \Pr[\mathsf{E}_1] - \frac{\ell_m+\ell_r}{2}\sqrt{\left(1 + 2^{-r}\right)^d - 1}$.

Proof. The result follows directly from the ring version of the leftover hash lemma [BJRW20, Lemma 7]. □

Game_3: In this game, we run $\mathcal{O}_{\mathsf{pre}}$ as $\mathcal{O}_1$ defined in Figure 5.

Lemma 3.5. $\Pr[\mathsf{E}_3] = \Pr[\mathsf{E}_2]$.

Proof. Define CTR to be the random variable which is the value of ctr when $\mathcal{O}_1$ actually outputs $(x,\mathbf{s})$. Then, for any $(\mathbf{m},\mathbf{r})$ such that $\|(\mathbf{m},\mathbf{r})\| \leq \mathcal{B}_m$, and any possible output $(x,\mathbf{s})$ we have:

$$\begin{aligned}\Pr[(x,\mathbf{s}) \leftarrow \mathcal{O}_1(\mathbf{m},\mathbf{r})] &= \sum_{i=0}^{\infty} \Pr[(x,\mathbf{s}) \leftarrow \mathcal{O}_1(\mathbf{m},\mathbf{r}) \wedge \mathsf{CTR} = i] \\ &= \sum_{i=0}^{\infty} \frac{1}{M}\left(1 - \frac{1}{M}\right)^i \Pr[(x,\mathbf{s}) \leftarrow \mathcal{O}_0(\mathbf{m},\mathbf{r})] \\ &= \Pr[(x,\mathbf{s}) \leftarrow \mathcal{O}_0(\mathbf{m},\mathbf{r})].\end{aligned}$$

When $\|(\mathbf{m},\mathbf{r})\| > \mathcal{B}_m$, then both $\mathcal{O}_1$ and $\mathcal{O}_0$ output $\perp$ in the first step. Hence, the two oracles behave identically and thus $\Pr[\mathsf{E}_3] = \Pr[\mathsf{E}_2]$. □

Game_4: In this game, we run $\mathcal{O}_{\mathsf{pre}}$ as $\mathcal{O}_2$ defined in Figure 5.

Lemma 3.6. $\Pr[\mathsf{E}_4] \geq \Pr[\mathsf{E}_3] - Q2^{-\lambda}$.

Proof. The only change is that we abort when ctr reaches $T_{\max}$. Hence, the statistical distance between $\mathcal{O}_2(\mathbf{m},\mathbf{r})$ and $\mathcal{O}_1(\mathbf{m},\mathbf{r})$ is at most $\left(1-\frac{1}{M}\right)^{T_{\max}} \leq 2^{-\lambda}$. Hence,

$$\Pr[\mathsf{E}_4] \geq \Pr[\mathsf{E}_3] - Q\left(1-\frac{1}{M}\right)^{T_{\max}} \geq \Pr[\mathsf{E}_3] - Q2^{-\lambda}.$$

□

Game_5: It differs from the previous game in a sense that we abort whenever $\mathcal{O}_2$ picks an index x, which was already sampled, even if it was not sent to the adversary. Recall that we still use the preimage oracle $\mathcal{O}_2$.

Lemma 3.7. $\Pr[\mathsf{E}_5] \geq \Pr[\mathsf{E}_4] - \frac{T_{\max}^2 Q^2}{2N}$.

Proof. The inequality follows directly from the birthday bound. □

Game_6: We still use the preimage oracle $\mathcal{O}_2$. The only change is in the challenger execution. First, the challenger additionally keeps track of all generated variables. Namely, define

$$\mathcal{X} := \left\{ \left(x, \mathbf{s} - \mathbf{D}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}\right) : x \text{ and } \mathbf{s} \text{ were generated while querying } \mathcal{O}_2(\mathbf{m},\mathbf{r}) \right\}. \tag{8}$$

In particular, $\mathcal{X}$ contains pairs which were not sent to adversary. Further,

$$\mathcal{Y} := \left\{ \left(x, \mathbf{s} - \mathbf{D}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}\right) : x \text{ and } \mathbf{s} \text{ were sent to } \mathcal{A} \text{ by } \mathcal{O}_2(\mathbf{m},\mathbf{r}) \right\}. \tag{9}$$

Then, we have the following simple relations: $|\mathcal{X}| = T_{\max}Q, |\mathcal{Y}| = |\mathcal{M}| \leq Q$ and $\mathcal{Y} \subseteq \mathcal{X}$. In this game, the challenger aborts when the output vectors from $\mathcal{A}$ satisfy the following:

$$\left(x^*, \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}\right) \in \mathcal{X}\backslash\mathcal{Y}.$$

Lemma 3.8. $\Pr[\mathsf{E}_6] \geq \frac{1}{2}\cdot\Pr[\mathsf{E}_5]$.

Proof. Denote $\mathcal{X}_0 := \{x \in [N] : \exists \mathbf{s}, (x,\mathbf{s}) \in \mathcal{X}\}$ and similarly for $\mathcal{Y}_0$. It is easy to see that

$$\Pr[\mathsf{E}_6 \wedge x^* \in \mathcal{Y}_0] = \Pr[\mathsf{E}_5 \wedge x^* \in \mathcal{Y}_0] \quad \text{and} \quad \Pr[\mathsf{E}_6 \wedge x^* \notin \mathcal{X}_0] = \Pr[\mathsf{E}_5 \wedge x^* \notin \mathcal{X}_0].$$

In the following, we show $\Pr[\mathsf{E}_6 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0] \geq \frac{1}{2}\Pr[\mathsf{E}_5 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0]$ which concludes the proof. Suppose that $\mathcal{A}$ wins Game_5 and the output index is in $\mathcal{X}_0\backslash\mathcal{Y}_0$, i.e. there exists a unique $\mathbf{z}$ such that $(x^*,\mathbf{z}) \in \mathcal{X}\backslash\mathcal{Y}$ (by conditions of Game_5). Because this pair does not belong to $\mathcal{Y}$, it was rejected during the execution of $\mathcal{O}_2$ on some input $(\mathbf{m}',\mathbf{r}')$. Denote

$$\mathbf{z}^* := \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^* - \mathbf{m}'\\ \mathbf{r}^* - \mathbf{r}'\end{bmatrix}.$$

By definition of $\mathcal{X}\backslash\mathcal{Y}$, $\mathbf{s}' := \mathbf{z} + \mathbf{D}\begin{bmatrix}\mathbf{m}'\\ \mathbf{r}'\end{bmatrix}$ is perfectly hidden among the preimages of $f(x^*) + \mathbf{C}\begin{bmatrix}\mathbf{m}'\\ \mathbf{r}'\end{bmatrix}$ under $\mathbf{A}$ since the adversary did not actually see $\mathbf{s}'$. Let $\mathbf{u}$ be a polynomial vector with ternary coefficients in $\Lambda_q^{\perp}(\mathbf{A})$, which exists since $\mathsf{m} > \mathsf{n}\log q$. Note that

$$\|\mathbf{u}\| \leq \sqrt{\mathsf{m}d} \quad \text{and} \quad \mathfrak{s} \geq \alpha\mathcal{B}_m d\sqrt{(\ell_m+\ell_r)\mathsf{m}} \geq \sqrt{\mathsf{m}d} \geq \|\mathbf{u}\|.$$

Therefore,

$$\mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix} = \mathbf{z} \iff \mathbf{s}' = \mathbf{z}^*$$

and also

$$\begin{aligned}\Pr\left[\mathbf{s}' = \mathbf{z}^* \middle| \mathsf{E}_5 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0\right] &\leq \frac{\exp\left(\frac{-\|\mathbf{z}^*\|^2}{-2\mathfrak{s}^2}\right)}{\exp\left(\frac{-\|\mathbf{z}^*\|^2}{2\mathfrak{s}^2}\right) + \exp\left(\frac{-\|\mathbf{z}^*+\mathbf{u}\|^2}{2\mathfrak{s}^2}\right) + \exp\left(-\frac{\|\mathbf{z}^*+\mathbf{u}\|^2}{2\mathfrak{s}^2}\right)}\\ &\leq \frac{1}{1 + \left(\exp\left(-\frac{\langle \mathbf{z}^*,\mathbf{u}\rangle}{\mathfrak{s}^2}\right) + \exp\left(\frac{\langle \mathbf{z}^*,\mathbf{u}\rangle}{\mathfrak{s}^2}\right)\right)\exp\left(-\frac{\|\mathbf{u}\|^2}{2\mathfrak{s}^2}\right)}\\ &\leq \frac{1}{1 + 2\exp\left(-\frac{\|\mathbf{u}\|^2}{2\mathfrak{s}^2}\right)}\\ &\leq \frac{1}{1 + 2\exp\left(-\frac{1}{2}\right)} \leq \frac{1}{2}.\end{aligned}$$

By taking the complement, we deduce that

$$\Pr\left[\mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix} \neq \mathbf{z} \middle| \mathsf{E}_5 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0\right] \geq \frac{1}{2}.$$

Finally, note that if $\mathsf{E}_5 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0$ hold and additionally $\mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix} \neq \mathbf{z}$, then we must have

$$\left(x^*, \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}\right) \notin \mathcal{X}\backslash\mathcal{Y}$$

since $(x^*, \mathbf{z}) \in \mathcal{X}\backslash\mathcal{Y}$ and there is at most one pair in $\mathcal{X}$ of the form $(x^*, \cdot)$ by the condition introduced in Game_5. This implies that

$$\Pr[\mathsf{E}_6 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0] \geq \frac{1}{2}\Pr[\mathsf{E}_5 \wedge x^* \in \mathcal{X}_0\backslash\mathcal{Y}_0]$$

which concludes the proof. □

Game_7: At the beginning, the challenger samples an index $j^* \leftarrow [Q]$. Then at the end, if $x^* \in \mathcal{Y}_0$ and x^* was *not* sampled in the j^*-th query then the challenger aborts. We still use the preimage oracle $\mathcal{O}_2$.

Lemma 3.9. $\Pr[\mathsf{E}_7] \geq \frac{1}{Q}\Pr[\mathsf{E}_6]$.

Proof. Clearly, if $x^* \notin \mathcal{Y}_0$ then the game behaves identically as before. However, if $x^* \in \mathcal{Y}$ then with probability at least $1/Q$ we have that x^* was sampled during the j^*-th oracle query. □

Game_8: In this game, only for the j^*-th oracle query, instead of executing $\mathcal{O}_2$ we run $\mathcal{O}_3$ as in Figure 5.

Lemma 3.10. $\Pr[\mathsf{E}_8] \geq \Pr[\mathsf{E}_7] - T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - 2^d$.

Proof. First, we exclude the case that $\eta'_\epsilon(\Lambda_q^\perp(\mathbf{A})) > \eta_{\min}$ which occurs with probability 2^{-d}. Then, by our parameter selection and Lemma 2.5, the statistical distance between $\mathcal{O}_3(\mathbf{m}, \mathbf{r})$ and $\mathcal{O}_2(\mathbf{m}, \mathbf{r})$ is at most $\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}$. Hence, the statement follows by the hybrid argument. □

Game_9: For presentation purposes, we summarise the security game in Figure 6. The difference from the previous game is that the adversary loses whenever

$$\left(x^*, \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}\right) \in \mathcal{Y}.$$

Game_9

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{\mathsf{n}\times\mathsf{m}}$.
2. Sample $\mathbf{D} \leftarrow \{0,1\}^{\mathsf{m}d\times(\ell_m+\ell_r)d}$ and set $\mathbf{C} = \mathbf{AD}$.
3. Set $\mathcal{M} = \emptyset$.
4. Sample $j^* \leftarrow [Q]$.
5. Run $(x^*, \mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*) \leftarrow \mathcal{A}^{\mathcal{O}'}(\mathbf{A}, \mathbf{C})$, where $\mathcal{O}'$ is defined below.
6. Define $\tilde{\mathbf{s}} = \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}$
7. If all the conditions below hold, then return 1 and 0 otherwise:
   - $\mathbf{m}^* \notin \mathcal{M}$ and $\|(\mathbf{m}^*, \mathbf{r}^*)\| \le \mathcal{B}_m$ and $\|\mathbf{s}^*\| \le \mathcal{B}_s$ (from the $\mathsf{Int\text{-}ISIS}_f$ game),
   - $\mathbf{A}\tilde{\mathbf{s}} = f(x^*)$ (from the $\mathsf{Int\text{-}ISIS}_f$ game),
   - $\forall (x,\mathbf{s}) \neq (x',\mathbf{s}') \in \mathcal{X}$, $x \neq x'$ (from Game_5),
   - $(x^*, \tilde{\mathbf{s}}) \notin \mathcal{X}\backslash\mathcal{Y}$ (from Game_6),
   - if $x^* \in \mathcal{Y}_0$, then x^* must have been generated in the j^*-th oracle query (from Game_7),
   - $(x^*, \tilde{\mathbf{s}}) \notin \mathcal{Y}$ (from Game_9).

Fig. 6: Description of Game_9. Here, $\mathcal{O}'$ behaves as $\mathcal{O}_2$, apart from the j^*-th query where it behaves like $\mathcal{O}_3$ (defined in Figure 5.)

Lemma 3.11. $\Pr[\mathsf{E}_9] \ge \frac{1}{3}\Pr[\mathsf{E}_8] - \frac{\ell_m \cdot 2^{-rd}}{3}$.

Proof. We follow the proof strategy from [LM13, Theorem 3.2]. Note that conditioned on $x^* \notin \mathcal{Y}_0$ the two security games are identical. Hence, in the following we show

$$\Pr[\mathsf{E}_9 \wedge x^* \in \mathcal{Y}_0] \ge \frac{1}{3}\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0] - \frac{\ell_m \cdot 2^{-rd}}{3}.$$

Let us consider the following alternative game Game'_9 defined in Figure 7 and denote E'_9 to be the event that the adversary wins Game'_9. We claim that

$$\Pr[\mathsf{E}'_9] = \Pr[\mathsf{E}_9 \wedge x^* \in \mathcal{Y}_0].$$

Indeed, the only change is that the challenger *resamples* the vectors $\bar{\mathbf{s}}_{j^*}$ (which is the preimage generated in the j^*-th query) and $\mathbf{D}$ according to their original distributions conditioned on what the adversary $\mathcal{A}$ already knows (e.g. it knows the value $\mathbf{s}_j$). Hence, from now on we focus on Game'_9.

Let us denote $\tilde{\mathbf{s}}_{j^*} := \mathbf{s}^* - \mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}$. In the following, we will use the fact that when $b = 0$, we have $\tilde{\mathbf{s}}_{j^*} = \tilde{\mathbf{s}}$. Next, we partition the success probability into two parts: $\Pr[\mathsf{E}'_9] = \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} \neq \mathbf{s}_j] + \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \mathbf{s}_j]$. Now, one observes that

$$\begin{aligned}
\Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} \neq \mathbf{s}_j] &\ge \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} \neq \bar{\mathbf{s}}_{j^*} \wedge b = 0]\\
&\ge \frac{1}{3}\cdot\Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} \neq \bar{\mathbf{s}}_{j^*} | b = 0]\\
&\ge \frac{1}{3}\Pr\left[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}} \neq \bar{\mathbf{s}} \wedge \tilde{\mathbf{s}}_{j^*} \neq \bar{\mathbf{s}}_{j^*} | b = 0\right]\\
&\ge \frac{1}{3}\Pr\left[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} \neq \bar{\mathbf{s}}_{j^*}\right].
\end{aligned}$$

For the second part, let us denote

$$S(\mathbf{D}, i^*) := \left\{\left[\mathbf{d}_1 \cdots \mathbf{d}_{i^*-1}\ \mathbf{e}\ \mathbf{d}_{i^*+1} \cdots \mathbf{d}_{\ell_m+\ell_r}\right] : \mathbf{Ae} = \mathbf{Ad}_{i^*} \wedge \mathbf{e} \in \{0,1\}^{(\ell_m+\ell_r)d}\right\}.$$

$\underline{\mathsf{Game}'_9}$

1. Generate $\mathbf{A} \leftarrow \mathcal{R}_q^{\mathsf{n}\times\mathsf{m}}$.
2. Sample $\mathbf{D} = (\mathbf{d}_i)_{i\in[\ell_m+\ell_r]} \leftarrow \{0,1\}^{\mathsf{m}d\times(\ell_m+\ell_r)d}$ and set $\mathbf{C} = \mathbf{AD}$.
3. Set $\mathcal{M} = \emptyset$.
4. Sample $j^* \leftarrow [Q]$.
5. Run $(x^*, \mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*) \leftarrow \mathcal{A}^{\mathcal{O}'}(\mathbf{A}, \mathbf{C})$, where $\mathcal{O}'$ as in Figure 6.
6. If one of the conditions below does not hold, then return 0:
   - $\mathbf{m}^* \notin \mathcal{M}$ and $\|(\mathbf{m}^*, \mathbf{r}^*)\| \le \mathcal{B}_m$ and $\|\mathbf{s}^*\| \le \mathcal{B}_s$ (from the $\mathsf{Int\text{-}ISIS}_f$ game),
   - $\forall (x, \mathbf{s}) \neq (x', \mathbf{s}') \in \mathcal{X}$, $x \neq x'$ (from Game_5),
   - x^* must have been generated in the j^*-th oracle query (from Game_7 and the condition $x^* \in \mathcal{Y}_0$).
7. Let $(\mathbf{m}_{j^*}, \mathbf{r}_{j^*})$ be the input for the j^*-th query, $\bar{\mathbf{s}}_{j^*}$ be the generated preimage, and denote the query output as $(x^*, \mathbf{s}_{j^*}) \in \mathcal{Y}$.
8. Let $i^* \in [\ell_m]$ be the index for which $m^*_{i^*} \neq m_{j^*,i^*}$.
9. Flip a bit $b \leftarrow D$, where D is defined as $D(0) = 1/3$ and $D(1) = 2/3$.
10. If $b = 0$, then set $\bar{\mathbf{s}} = \bar{\mathbf{s}}_{j^*}$ and $\bar{\mathbf{D}} = \mathbf{D}$.
11. If $b = 1$, sample $\bar{\mathbf{s}} \leftarrow D_{\mathsf{s}}^{\mathsf{m}d}$ and $\bar{\mathbf{d}} \leftarrow \{0,1\}^{\mathsf{m}d}$ conditioned on

$$\mathbf{A}\bar{\mathbf{s}} = f(x^*), \quad \mathbf{A}\bar{\mathbf{d}} = \mathbf{A}\mathbf{d}_{i^*}, \quad \mathbf{s}_{j^*} = \bar{\mathbf{s}} + \bar{\mathbf{D}} \begin{bmatrix} \mathbf{m}_{j^*} \\ \mathbf{r}_{j^*} \end{bmatrix}$$

   where $\bar{\mathbf{D}} := \left[\mathbf{d}_1 \cdots \mathbf{d}_{i^*-1}\ \bar{\mathbf{d}}\ \mathbf{d}_{i^*+1} \cdots \mathbf{d}_{\ell_m+\ell_r}\right]$.
12. Define $\tilde{\mathbf{s}} = \mathbf{s}^* - \bar{\mathbf{D}} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{r}^* \end{bmatrix}$
13. If all the conditions below hold, then return 1 and 0 otherwise:
   - $\mathbf{A}\tilde{\mathbf{s}} = f(x^*)$ (from the $\mathsf{Int\text{-}ISIS}_f$ game),
   - $\tilde{\mathbf{s}} \neq \bar{\mathbf{s}}$ (from Game_9 and the condition that $x^* \in \mathcal{Y}_0$).

Fig. 7: Description of the alternative game Game'_9.

Clearly, $\mathbf{D} \in S(\mathbf{D}, i^*)$. Further,

$$\begin{aligned}\Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \mathbf{s}_j] &\ge \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| \ge 2] \\ &\ge \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| \ge 2 \wedge b = 1].\end{aligned}$$

For presentation purposes, define the event

$$E := (\mathsf{E}_8) \wedge (x^* \in \mathcal{Y}_0) \wedge (\tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}) \wedge (|S(\mathbf{D}, i^*)| \ge 2) \wedge (b = 1).$$

Then,

$$\begin{aligned}\Pr[E] &\ge \frac{2}{3}\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| \ge 2] \\ &\ge \frac{2}{3}\left(\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] - \Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| = 1]\right) \\ &\ge \frac{2}{3}\left(\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] - \ell_m\left(\frac{q^{\mathsf{n}}}{2^{\mathsf{m}}}\right)^d\right)\end{aligned}$$

where for the last inequality we used the fact from [Lyu12, Lemma 5.2] to deduce that

$$\begin{aligned}\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| = 1] &\le \Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge \exists i, |S(\mathbf{D}, i)| = 1] \\ &\le \Pr[\exists i, |S(\mathbf{D}, i)| = 1] \\ &\le \sum_{i=1}^{\ell_m} \Pr[|S(\mathbf{D}, i)| = 1]\end{aligned}$$

$$\leq \ell_m \left(\frac{q^{\mathsf{n}}}{2^{\mathsf{m}}}\right)^d = \ell_m \cdot 2^{-rd}.$$

Thus,

$$\begin{aligned}\Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] &\geq \Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*} \wedge |S(\mathbf{D}, i^*)| \geq 2 \wedge b = 1] \\ &\geq \Pr[\tilde{\mathbf{s}} \neq \bar{\mathbf{s}} | E] \cdot \Pr[E].\end{aligned}$$

We claim that

$$\Pr[\tilde{\mathbf{s}} = \bar{\mathbf{s}} | E] \leq \Pr[\bar{\mathbf{s}} = \bar{\mathbf{s}}_{j^*} \wedge \bar{\mathbf{D}} = \mathbf{D} | E].$$

Indeed, suppose $\tilde{\mathbf{s}} = \bar{\mathbf{s}}$ which by definition implies

$$\mathbf{s}^* = \bar{\mathbf{s}} + \bar{\mathbf{D}} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{r}^* \end{bmatrix}.$$

We know from E that $\tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}$, i.e.

$$\bar{\mathbf{s}}_{j^*} + \mathbf{D} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{r}^* \end{bmatrix} = \mathbf{s}^* = \bar{\mathbf{s}} + \bar{\mathbf{D}} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{r}^* \end{bmatrix}.$$

Also, we know from Step 11 of Figure 7 that:

$$\bar{\mathbf{s}}_{j^*} + \mathbf{D} \begin{bmatrix} \mathbf{m}_{j^*} \\ \mathbf{r}_{j^*} \end{bmatrix} = \mathbf{s}_{j^*} = \bar{\mathbf{s}} + \bar{\mathbf{D}} \begin{bmatrix} \mathbf{m}_{j^*} \\ \mathbf{r}_{j^*} \end{bmatrix}.$$

By subtracting the two equations, we end up with

$$(\mathbf{D} - \bar{\mathbf{D}}) \begin{bmatrix} \mathbf{m}^* - \mathbf{m}_{j^*} \\ \mathbf{r}^* - \mathbf{r}_{j^*} \end{bmatrix} = \mathbf{0}.$$

Since both matrices $\mathbf{D}, \bar{\mathbf{D}}$ belong to $S(\mathbf{D}, i^*)$, we have

$$(m^*_{i^*} - m_{j^*,i^*})(\mathbf{d}_{j,i^*} - \bar{\mathbf{d}}_{i^*}) = \mathbf{0}$$

where $\mathbf{d}_{i^*}, \bar{\mathbf{d}}_{i^*}$ are the i^*-th columns of $\mathbf{D}, \bar{\mathbf{D}}$ respectively. Since $2\mathcal{B}_m < q$, we deduce that this equation holds over integers. Further, we know that $m^*_{i^*} \neq m_{j^*,i^*}$ which implies that $\mathbf{d}_{i^*} = \bar{\mathbf{d}}_{i^*}$ and thus $\mathbf{D} = \bar{\mathbf{D}}$. Consequently, $\bar{\mathbf{s}} = \bar{\mathbf{s}}_{j^*}$ and the claim follows. Hence, we have

$$\Pr[\tilde{\mathbf{s}} = \bar{\mathbf{s}} | E] \leq \Pr[\bar{\mathbf{s}} = \bar{\mathbf{s}}_{j^*} \wedge \bar{\mathbf{D}} = \mathbf{D} | E] \leq \Pr[\bar{\mathbf{D}} = \mathbf{D} | E] \leq \frac{1}{2}.$$

This is because E includes the event that $S(\mathbf{D}, i^*)$ contains at least two elements. Therefore, we conclude that

$$\begin{aligned}\Pr[\mathsf{E}'_9 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] &\geq \frac{1}{2} \cdot \frac{2}{3} \left(\Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] - 2^{-rd}\right) \\ &\geq \frac{1}{3} \Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0 \wedge \tilde{\mathbf{s}}_{j^*} = \bar{\mathbf{s}}_{j^*}] - \frac{\ell_m \cdot 2^{-rd}}{3}\end{aligned}$$

and thus

$$\Pr[\mathsf{E}_9 \wedge x^* \in \mathcal{Y}_0] \geq \Pr[\mathsf{E}'_9] \geq \frac{1}{3} \Pr[\mathsf{E}_8 \wedge x^* \in \mathcal{Y}_0] - \frac{\ell_m \cdot 2^{-rd}}{3}$$

which finishes the proof. □

Game_{10}: In this game, throughout the experiment $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{pre}}$ defined as $\mathcal{O}_3$ in Figure 5 (not only for the j^*-th query).

Lemma 3.12. $\Pr[\mathsf{E}_{10}] \geq \Pr[\mathsf{E}_9] - (Q-1)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - 2^{-d}$.

Proof. It follows identically as in Lemma 3.10. □

Finally, our reduction runs as a challenger in Game_{10}. Namely, it first gets $T_{\max}Q$ pairs $(x_i, \mathbf{s}_i)_{i\in[T_{\max}Q]}$ from the ISIS_f challenger, i.e. $\mathbf{As} = f(x_i)$, and then uses them to simulate the query outputs from $\mathcal{O}_3$. Note that if $\mathcal{A}$ wins Game_{10} then the set $\mathcal{X}$ contains exactly these pairs, and further the output $(x^*, \tilde{\mathbf{s}})$ does not belong to $\mathcal{X}$ (from Game_6 and Game_9). Also,

$$\|\tilde{\mathbf{s}}\| \leq \|\mathbf{s}^*\| + \left\|\mathbf{D}\begin{bmatrix}\mathbf{m}^*\\ \mathbf{r}^*\end{bmatrix}\right\| \leq \mathcal{B}_s + \mathcal{B}_m d\sqrt{(\ell_m+\ell_r)\mathsf{m}} = \mathcal{B}.$$

Therefore, the reduction outputs a valid ISIS_f solution. Now, using all the previous lemmas, the probability that the reduction wins the ISIS_f game can be lower-bounded as

$$\begin{aligned}
\Pr[\mathsf{E}_{10}] &\geq \Pr[\mathsf{E}_9] - (Q-1)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - 2^{-d}\\
&\geq \frac{1}{3}\Pr[\mathsf{E}_8] - \frac{\ell_m\cdot 2^{-rd}}{3} - (Q-1)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - 2^{-d}\\
&\geq \frac{1}{3}\Pr[\mathsf{E}_7] - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{3Q}\Pr[\mathsf{E}_6] - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{6Q}\Pr[\mathsf{E}_5] - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{6Q}\Pr[\mathsf{E}_4] - \frac{T_{\max}^2 Q}{12N} - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{6Q}\Pr[\mathsf{E}_3] - \frac{2^{-\lambda}}{6} - \frac{T_{\max}^2 Q}{12N} - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{6Q}\Pr[\mathsf{E}_2] - \frac{2^{-\lambda}}{6} - \frac{T_{\max}^2 Q}{12N} - \frac{\ell_m\cdot 2^{-rd}}{3} - \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) - \frac{2^{-d+2}}{3}\\
&\geq \frac{1}{6Q}\Pr[\mathsf{E}_1] - \mathsf{AddLoss}
\end{aligned}$$

where

$$\begin{aligned}
\mathsf{AddLoss} := \frac{\ell_m+\ell_r}{12Q}\sqrt{\left(1+2^{-r}\right)^d - 1} + \frac{2^{-\lambda}}{6} + \frac{T_{\max}^2 Q}{12N} + \frac{\ell_m\cdot 2^{-rd}}{3}\\
+ \left(Q-\frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M}\right) + \frac{2^{-d+2}}{3}
\end{aligned}$$

which concludes the proof. □

Asymptotic parameter selection. Let $d = O(\lambda)$, $\alpha = O(\sqrt{\lambda})$ and $\epsilon = \mathsf{negl}(\lambda)$. Then, $M = O(1), \varepsilon = \mathsf{negl}(\lambda)$ and $T_{\max} = O(\lambda)$.

Suppose we want $\mathsf{Int\text{-}ISIS}_f$ to be hard against efficient adversaries, and thus $Q = \mathsf{poly}(\lambda)$. This implies that we need N to be exponential in λ. Next, to use the leftover hash lemma, we set $r \geq \omega(\log d)$. With these parameters, if a PPT adversary $\mathcal{A}$ solves $\mathsf{Int\text{-}ISIS}_f$ with probability δ, then there is a PPT adversary which solves ISIS_f with probability $\delta/(12Q) - \mathsf{negl}(\lambda)$.

3.3 Applications to Exotic Signatures

We provide a brief intuition on how to build digital signatures [GPV08; PFH+17], group signatures [BMW03; PLS18] and blind signatures [Fis06; AKSY22; PK22] from the (interactive) ISIS_f assumption.

Signature schemes. We can directly build standard model signatures using the hash-and-sign GPV framework. Namely, the secret key is a trapdoor for the public matrix $\mathbf{A}$. Let $\mathbf{C}$ be another uniformly random matrix (which corresponds to the $\mathbf{C}$ in the interactive ISIS_f problem). Then, to sign a (short) message $\mathbf{m}$, we sample a uniformly random $x \leftarrow [N]$ and compute a short preimage $\mathbf{s}$ which satisfies $\mathbf{As} = f(x) + \mathbf{Cm}$. Thus, the signature is a pair $(x, \mathbf{s})$. Unforgeability property follows directly from the $\mathsf{Int\text{-}ISIS}_f$ assumption.

Group signatures. Let L be the size of the group and for $i \in [L]$, denote $\mathbf{m}_i$ to be the polynomial vector for which its coefficient vector is a binary representation of i. We follow the standard sign-and-encrypt approach. Namely, the setup authority generates matrices $(\mathbf{A}, \mathbf{C})$, along with the trapdoor for preimage sampling. Then, for each user $i \in [L]$, it generates a uniformly random $x_i \leftarrow [N]$ and computes a short preimage $\mathbf{s}_i$ that satisfies $\mathbf{As}_i = f(x_i) + \mathbf{Cm}_i$. Hence, the signature for user i consists of a zero-knowledge proof of knowledge π of $i \in [L], x_i \in [N]$, and short $\mathbf{s}_i$ such that $\mathbf{As}_i = f(x_i) + \mathbf{Cm}_i$ [12]. Since we obtain π by applying the Fiat-Shamir transformation to an interactive proof, the message to sign is included as an input to the hash function.

Blind signatures. We follow the Fischlin framework for constructing round-optimal blind signatures. The public key are the matrices $(\mathbf{A}, \mathbf{C})$, while the secret key is a trapdoor for $\mathbf{A}$. Suppose the user wants to obtain a signature on a short message vector $\mathbf{m}$. It first computes the Ajtai commitment [Ajt96] on $\mathbf{m}$, i.e. $\mathbf{u} := \mathbf{C}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix}$, where $\mathbf{r}$ is a fresh, short randomness vector, and sends $\mathbf{u}$ to the signer along with a proof π_u of well-formedness of $\mathbf{u}$ .The signer, who possess the trapdoor for matrix $\mathbf{A}$, samples $x \leftarrow [N]$ and returns to the user a short vector $\mathbf{s}$ which satisfies $\mathbf{As} = f(x) + \mathbf{u}$. Finally, the user outputs as a signature a zero-knowledge proof of knowledge π_s of $x \in [N]$, short randomness $\mathbf{r}$ and a short vector $\mathbf{s}$ such that

$$\mathbf{As} = f(x) + \mathbf{C}\begin{bmatrix}\mathbf{m}\\ \mathbf{r}\end{bmatrix},$$

where $\mathbf{m}$ is a part of the statement. This is also the approach we take for constructing anonymous credentials in Section 8 (see Section 4 for more background on anonymous credentials).

4 Background on Anonymous Credentials

We recall the definitions of the anonymous credentials. The entities involved in the scheme are the following: (i) the issuer $\mathcal{I}$ certifies attributes of holders, (ii) the holder $\mathcal{H}$ receives credentials about their attributes and uses them to authenticate to third parties, and (iii) the verifier $\mathcal{V}$ engages in a protocol with a holder $\mathcal{H}$, and learns a subset of the attributes of the holder.

Definition 4.1 (Anonymous Credentials). *An anonymous credentials scheme is a triple* $(\mathsf{Init}, \mathsf{Issue}_{\mathcal{I},\mathcal{H}}, \mathsf{Verify}_{\mathcal{H},\mathcal{V}})$ *of algorithms defined as follows:*

- $(\mathsf{ipk}, \mathsf{isk}) \leftarrow \mathsf{Init}(1^\lambda, 1^l)$*: the initialisation algorithm takes as input the security parameter* 1^λ *and the nonzero size* l *(in unary) of the array of attributes the issuer will certify, and generates the issuer public key* ipk *and the issuer secret key* isk*.*
- $\mathsf{cred}/\perp \leftarrow \mathsf{Issue}_{\mathcal{I},\mathcal{H}}\langle(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}), (\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx})\rangle$*: this is the issuing protocol between issuer* $\mathcal{I}$ *and holder* $\mathcal{H}$*. The holder has as input the issuer public key* ipk*, an array* $\mathsf{attrs} := [\mathsf{a_i}]_{i\in[l]}$ *of attributes* $\mathsf{a_i} \in \{0,1\}^*$ *and an array* $\mathsf{idx} \in [l]^{l'}, 1 \le l' \le l$ *of* l' *pairwise distinct indices for the attributes in* attrs *that will be revealed to the issuer. The issuer has as input the issuer secret key* isk*, an array of disclosed attributes* $\mathsf{attrs}' := [\mathsf{a}'_i]_{i\in\mathsf{idx}}$ *and indices* idx*. The holder outputs a credential* cred *over attributes* attrs *if* $\forall i \in \mathsf{idx}, \mathsf{attrs}'_i = \mathsf{attrs}_i$ *or* $\perp$ *otherwise.*

[12] The signature should also contain a verifiable encryption of i using the opener's public key.

– $\langle \perp, 1/0 \rangle \leftarrow \mathsf{Verify}_{\mathcal{H},\mathcal{V}}\langle(\mathsf{cred}, \mathsf{attrs}, \mathsf{idx}), (\mathsf{ipk}, \mathsf{attrs}', \mathsf{idx})\rangle$*: this is a protocol between a holder* $\mathcal{H}$ *and a verifier* $\mathcal{V}$*. The holder* $\mathcal{H}$ *has as input a credential* cred*, an array* $\mathsf{attrs} := [\mathsf{a_i}]_{i\in[l]}$ *of attributes* $\mathsf{a_i} \in \{0,1\}^*$ *and an array* $\mathsf{idx} \in [l]^{l'}, 1 \leq l' \leq l$ *of* l' *pairwise distinct indices for the attributes in* attrs *that will be revealed to the verifier. The verifier* $\mathcal{V}$ *has as input the issuer public key* ipk *of issuer* $\mathcal{I}$*, an array of disclosed attributes* $\mathsf{attrs}' := [\mathsf{a}'_i]_{i\in\mathsf{idx}}$ *and indices* idx*. The verifier outputs 1 if* cred *is a valid credential from issuer* $\mathcal{I}$ *over attributes* attrs *such that* $\forall i \in \mathsf{idx}, \mathsf{attrs}'_i = \mathsf{attrs}_i$*, and 0 otherwise.*

Related work. Anonymous credential systems offer to individuals the means to achieve control over the way information about themselves is exchanged for purposes such as authentication to third parties and payments. Camenisch and Lysyanskaya [CL01] present one of the first formal definitions and practical constructions for anonymous credential systems. The protocols that realise these schemes enable holders to receive credentials over their attributes from issuers, and later present subsets of these attributes to verifiers. These interactions preserve the privacy of holders upon issuance (since the holder can hide a subset of the certified attributes) and upon verification (since verifications are unlinkable and support the disclosure of subsets of the certified attributes). Subsequent constructions based on CL signature scheme [CL02a], BBS+ signature scheme [BBS04; ASM08] and associated efficient proof system [CDL16], or on signatures based on SXDH assumption [LMPY16] are integrated into prominent open-source self-sovereign identity projects such as Hyperledger Indy [Hypb], Hyperledger Aries [Hypa] or the MATTR stack [MATa], or are proposed as an IETF standard draft [LKWL22]. These more recent constructions are based on a combination of signature schemes supporting blind signing (to achieve issuer anonymity) and multi-message (to efficiently accumulate multiple attributes into a single signature); commitment schemes and efficient zero-knowledge proof systems (to achieve unlinkable presentations and prove properties about pairs of attributes).

Additional aspects. The definitions above are minimalistic: while they capture the essential features of anonymous credential systems (blind issuance for a subset of attributes, unlinkability of verifications and selective disclosure of attributes), they exclude several aspects that feature in prominent related works, whose absence we review and justify here. The first aspect relates to pseudonyms [Cha85; Che95; Dam88; LRSW99], which are optional, verifier-specific identifiers that a holder can present upon verification. Pseudonyms enable selective tracking of holders, since verifiers can link verifications that used the same pseudonym. Pseudonyms at different verifiers cannot be linked together, and verifier-specific pseudonyms cannot be forged arbitrarily by holders to fraudulently present different personas. Pseudonyms are typically realised by adding a user initialisation and registration phase during which the user generates a private key and has this key blindly signed by issuers; pseudonyms can then be derived by committing the private key in a non-hiding way against a verifier-specific base and proving equality between the signed private key and the committed one. While neither user private keys nor pseudonyms are captured in our definitions, we remark that these can be easily integrated on top of any scheme that instantiates our definition: user secret key may simply be one of the undisclosed attributes upon issuance; and verification may be extended to include commitments to undisclosed attributes alongside proofs of equality between committed and signed attributes to achieve pseudonyms. A closely related aspect is that of all-or-nothing transferability, that requires that sharing a credential amongst colluding holders require sharing private key material, which represents a strong incentive not to engage in such practices. This is once again typically achieved by including private keys in the issuance, whose knowledge must be proved upon verification. Such keys might further be stored in HSMs or similar hardware facilities to make the sharing of these keys impossible. Another aspect relates to enforceable one-show credentials [CL01]. This property has meanwhile been decoupled from identity system proper, and has been further developed in payment-systems under the wider notion of double-spending resistance. Finally, the topic of revocation of anonymous credentials has received a lot of attention. The approaches proposed in the literature [CL02b; CKS09; CKS10] can be typically integrated on top of a basic scheme, provided that it supports blind issuance and proof of knowledge of equality of committed attributes. These can be simply integrated on top of our construction, as it shall become apparent later.

Security Definitions. Informally, we require the scheme to be secure against any coalition comprising a malicious issuer, malicious holders and verifiers who attempt to learn more information about a victim

holder (e.g. learn an undisclosed attribute upon verification) or link the same victim holder across multiple verifications. For example, if a holder repeatedly authenticates disclosing only their vaccination status, no coalition of malicious verifiers/holders should learn any information about any of the other certified attributes, nor should they be able to learn whether the same holder or two different ones disclosed twice that same vaccination status. We also require credentials to be unforgeable, constraining the ability of any coalition of malicious holders to successfully pass verification against an honest verifier for an array of attributes different from those that were certified by an honest issuer.

In what follows we present definitions of *correctness*, *anonymity* and *one-more unforgeability*.

Definition 4.2 (Correctness). An anonymous credential scheme is correct if for any $\lambda \in \mathbb{N}$, $l \in \mathbb{N}$, $l' \in [l]$, $l'' \in [l]$, any $\mathsf{idx}' \in [l]^{l'}$ such that its elements are pairwise distinct, any $\mathsf{idx}'' \in [l]^{l''}$ such that its elements are pairwise distinct, any array $\mathsf{attrs} := [\mathsf{a}_0, \ldots, \mathsf{a}_{l-1}]$, $\mathsf{attrs}' := [\mathsf{a}'_0, \ldots, \mathsf{a}'_{l'-1}]$ and $\mathsf{attrs}'' := [\mathsf{a}''_0, \ldots, \mathsf{a}''_{l''-1}]$ of attributes such that $\forall i \in \mathsf{idx}', \mathsf{attrs}_i = \mathsf{attrs}'_i$ and $\forall i \in \mathsf{idx}'', \mathsf{attrs}_i = \mathsf{attrs}''_i$:

$$\Pr\left[(\mathsf{ipk}, \mathsf{isk}) \leftarrow \mathsf{Init}(1^\lambda, l) : \bot \leftarrow \mathsf{Issue}_{\mathcal{I},\mathcal{H}}\langle(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}'), (\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx}')\rangle\right] \leq \mathsf{negl}(\lambda) \tag{10}$$

and

$$\Pr\left[\begin{array}{c}(\mathsf{ipk}, \mathsf{isk}) \leftarrow \mathsf{Init}(1^\lambda, l),\\ \mathsf{cred} \leftarrow \mathsf{Issue}_{\mathcal{I},\mathcal{H}}\langle(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}'), (\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx}')\rangle\end{array} : \begin{array}{c}\langle\bot, 0\rangle \leftarrow\\ \mathsf{Verify}_{\mathcal{H},\mathcal{V}}\langle(\mathsf{cred}, \mathsf{attrs}, \mathsf{idx}''),\\ (\mathsf{ipk}, \mathsf{attrs}'', \mathsf{idx}'')\rangle\end{array}\right] \leq \mathsf{negl}(\lambda). \tag{11}$$

Definition 4.3 (Anonymity Game). The anonymity game is played between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

1. *Setup.* The adversary receives the security parameter 1^λ and outputs the issuer public key ipk.
2. *Challenge.* $\mathcal{A}$ chooses two arrays of attributes attrs_0 and attrs_1 such that $|\mathsf{attrs}_0| = l$ and $|\mathsf{attrs}_1| = l$, and an array of indices $\mathsf{idx}^* \in [l]^{l'}$, $1 \leq l' \leq l$ such that $\forall i \in \mathsf{idx}^*, \mathsf{attrs}_{0,i} = \mathsf{attrs}_{1,i}$. The challenger $\mathcal{C}$ picks $b \in \{0, 1\}$. Then, $\mathcal{A}$ and $\mathcal{C}$ run the $\mathsf{Issue}_{\mathcal{I},\mathcal{H}}\langle(\cdot, \cdot, \cdot, \cdot), (\mathsf{ipk}, \mathsf{attrs}_b, \mathsf{idx}^*)\rangle$ protocol, where the challenger plays the role of the holder on input attribute array attrs_b and revealed indices idx^* and the adversary plays the role of the issuer. If the protocol succeeds, the challenger stores the generated cred_b. Next, the adversary and the challenger run the $\mathsf{Verify}_{\mathcal{H},\mathcal{V}}\langle(\mathsf{cred}_b, \mathsf{attrs}_b, \mathsf{idx}^*), (\cdot, \cdot, \cdot)\rangle$ protocol, where the challenger plays the role of the holder on input the credential cred_b generated previously, attribute array attrs_b and revealed indices idx^* and the adversary plays the role of the verifier.
3. *Response.* $\mathcal{A}$ outputs b' and wins the game if $b' = b$.

Definition 4.4 (Anonymity). A credential scheme has the *Anonymity* property if no PPT adversary $\mathcal{A}$ can win the Anonymity game with advantage greater than $\frac{1}{2} + \mathsf{negl}(\lambda)$.

Definition 4.5 (One-More Unforgeability Game). The one-more unforgeability game is played between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

1. *Setup.* The challenger runs the Init algorithm, sends ipk to the adversary and stores isk and .
2. *Queries.* $\mathcal{A}$ has access to the following oracle:
   - $\mathcal{O}_{\mathsf{Issue}}(\mathcal{H}, \mathsf{attrs}, \mathsf{idx})$: adversary $\mathcal{A}$ requests the creation of a credential for holder $\mathcal{H}$ over disclosed attributes attrs at indices idx. Then, $\mathcal{C}$ and $\mathcal{A}$ run the $\mathsf{Issue}_{\mathcal{I},\mathcal{H}}\langle(\mathsf{isk}, \mathsf{attrs}, \mathsf{idx}), (\cdot, \cdot, \cdot)\rangle$ protocol where the challenger plays the role of the issuer $\mathcal{I}$ on input the issuer secret key isk, the requested attributes attrs and indices idx, and the adversary plays the role of the holder. If it succeeds, the issuer increments Q_{Issue}.
3. *Response.* We say that $\mathcal{A}$ wins the one-more unforgeability game if it outputs $\mathcal{L} := \{\langle\mathsf{attrs}_i, \mathsf{idx}_i\rangle\}$ of size $Q_{\mathsf{Issue}} + 1$ such that the following conditions hold:

– for all $i \in [Q_{\mathsf{Issue}} + 1]$, $\langle \perp, 1 \rangle \leftarrow \mathsf{Verify}_{\mathcal{H},\mathcal{V}}\langle(\cdot,\cdot,\cdot), (\mathsf{ipk}, \mathsf{attrs}_i, \mathsf{idx}_i)\rangle$ where the challenger plays the role of the verifier on input the issuer public key ipk and each of the pairs of attributes and indices $\langle \mathsf{attrs}_i, \mathsf{idx}_i \rangle \in \mathcal{L}$, and the adversary plays the role of the holder,
– for each distinct pairs $\langle \mathsf{attrs}_i, \mathsf{idx}_i \rangle \in \mathcal{L}$ and $\langle \mathsf{attrs}_j, \mathsf{idx}_j \rangle \in \mathcal{L}$, there exists at least one index i^* in both idx_i and idx_j such that $\mathsf{attrs}_{i,\mathsf{i}^*} \neq \mathsf{attrs}_{j,\mathsf{i}^*}$.

Definition 4.6 (One-More Unforgeability). A credential scheme is one-more unforgeable if no PPT adversary $\mathcal{A}$ can win the one-more unforgeability game with greater than negligible advantage.

5 Proof of Knowledge of a Preimage of $\mathsf{SIS}_{\mathsf{bin}}$

In the construction of anonymous credentials, we will need to provide a proof of knowledge of the $\mathsf{ISIS}_{\mathsf{bin}}$ secret, as in Equation 7. For the interactive $\mathsf{ISIS}_{\mathsf{bin}}$ case, it means proving knowledge of a vector $\vec{u} \in \mathbb{Z}_{\hat{q}}^{\mathsf{n}}$, as well as short vectors $\vec{s} \in \mathbb{Z}_{\hat{q}}^{\mathsf{m}}$ and $\vec{r} \in \mathbb{Z}_{\hat{q}}^{\ell_r}$ such that

$$\begin{aligned} &\vec{u} \in \{0,1\}^{\mathsf{t}}, \quad \|\vec{s}\| \leq \mathcal{B}_s \quad \text{and} \quad \|\vec{r}\| \leq \mathcal{B}_r, \\ &P\vec{s} = B\vec{u} + C \begin{bmatrix} \vec{m} \\ \vec{r} \end{bmatrix} \pmod{\hat{q}}, \end{aligned} \tag{12}$$

where

$$P \in \mathbb{Z}_{\hat{q}}^{\mathsf{n} \times \mathsf{m}}, \quad C \in \mathbb{Z}_{\hat{q}}^{\mathsf{n} \times (\ell_m + \ell_r)}, \quad \vec{m} \in \mathbb{Z}_{\hat{q}}^{\ell_m}, \quad B \in \mathbb{Z}_{\hat{q}}^{\mathsf{n} \times \mathsf{t}}, \quad \mathsf{Bounds} := (\mathcal{B}_s, \mathcal{B}_r)$$

are parts of the statement, along with some auxiliary information aux. Clearly, as long as $q | \hat{q}$, we can map equations such as (7) to this setting with standard transformation from $\mathcal{R}_q$ to $\mathbb{Z}_q^d$ using multiplication matrices (c.f. Section 2) and by lifting from $\mathbb{Z}_q$ to $\mathbb{Z}_{\hat{q}}$.

In order to prove (12), we make use of the LNP framework, developed recently by Lyubashevsky et al. [LNP22], for proving various lattice-related statements. For completeness, we provide the full protocol for proving (12) and refer to [LNP22] for more details.

5.1 Background

Notation. In the following, we introduce a few variables which might be also present in the other sections. We highlight that these are only defined for this section, and thus they have no relation with variables outside this section, unless specified otherwise.

Denote H to be the random oracle used for the NIZK. Let $\hat{d}$ be a power-of-two and $\hat{\mathcal{R}} := \mathbb{Z}[X]/(X^{\hat{d}} + 1)$. For a modulus $\hat{q} \in \mathbb{N}$, define $\hat{\mathcal{R}}_{\hat{q}} := \hat{\mathcal{R}}/(\hat{q})$. In this work, $\hat{q} = \hat{p}_1 \hat{p}_2$ is a product of two primes $\hat{p}_i$ of the form $\hat{p}_i = 5 \pmod 8$ and $\hat{p}_1 < \hat{p}_2$. For a polynomial $x \in \hat{\mathcal{R}}_{\hat{q}}$, we define $\tilde{x}$ to be the constant coefficient of x. Furthermore, we assume that $\hat{d}$ divides the security parameter λ.

Let $\sigma : \hat{\mathcal{R}}_{\hat{q}} \to \hat{\mathcal{R}}_{\hat{q}}$ be the ring automorphism defined by $\sigma(X) := X^{-1}$. For a polynomial vector (and similarly for a matrix) $\mathbf{x} = (x_1, \ldots, x_n) \in \hat{\mathcal{R}}_{\hat{q}}^n$, define $\sigma(\mathbf{x}) := (\sigma(x_1), \ldots, \sigma(x_n))$. Further, let $\tau \in \mathbb{N}$ be such that $1/\hat{p}_1^{\tau} = \mathsf{negl}(\lambda)$, i.e. a parameter used for soundness amplification.

ABDLOP Commitment. It was introduced in [LNP22] and extends both the Ajtai [Ajt96] and BDLOP [BDL+18] constructions. Let $n, m_1, m_2, \ell \in \mathbb{N}$ and $m_2 \geq n + \ell$. The commitment key consists of uniformly random matrices

$$\mathsf{crs} := (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}) \in \hat{\mathcal{R}}_{\hat{q}}^{n \times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n \times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{\ell \times m_2} \ .$$

Now, to commit to a message vector $\boldsymbol{s}_1 \in \hat{\mathcal{R}}_{\hat{q}}^{m_1}$ with small coefficients as well as a "full-fledged" polynomial vector $\boldsymbol{m} \in \hat{\mathcal{R}}_{\hat{q}}^{\ell}$, we sample a randomness vector $\boldsymbol{s}_2 \leftarrow \chi^{m_2}$, where χ is a probability distribution over $\hat{\mathcal{R}}_{\hat{q}}$, and compute:

$$\mathsf{ABDLOP.Com}(\mathsf{crs}, \boldsymbol{s}_1, \boldsymbol{m}; \boldsymbol{s}_2) := \begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{m} \end{bmatrix}.$$

Note that when $\ell = 0$ (resp. $m_1 = 0$) then this construction ends up being the Ajtai (resp. BDLOP) commitment scheme. In particular, the commitment size does not depend on the length m_1 of $\boldsymbol{s}_1$ (but it does on ℓ). Therefore, we will commit to long vectors with small coefficients in the "Ajtai" part $\boldsymbol{s}_1$ and commit to a few *garbage* polynomials used for the proofs in the "BDLOP" part $\boldsymbol{m}$. Hence, we call $\mathbf{t}_A$ (resp. $\mathbf{t}_B$) the "top part" (resp. "bottom part") of the commitment. An opening of the commitment is a triple $(\boldsymbol{s}_1, \boldsymbol{m}, \boldsymbol{s}_2)$[13].

Relaxed openings. As folklore in lattice-based cryptography, we also consider relaxed openings of a commitment which involve so-called *relaxation factors*. First, we fix $\xi, \nu > 0$ and a power-of-two k and define the following set $\mathcal{C}$ as[14]:

$$\mathcal{C} := \left\{ c \in \hat{\mathcal{R}}_{\hat{q}} : \|c\|_\infty \leq \xi \wedge \sigma_{-1}(c) = c \wedge \sqrt[2k]{\|c^{2k}\|_1} \leq \nu \right\}. \tag{13}$$

Roughly speaking, the first condition $\sigma_{-1}(c) = c$ is needed to prove quadratic equations in the committed messages which might additionally involve automorphisms. On the other hand, the second condition allows us to use [LNP22, Lemma 2.15] and deduce that if $\|\mathbf{r}\| \leq \mathcal{B}$ and $c \in \mathcal{C}$ then $\|c\mathbf{r}\| \leq \nu\mathcal{B}$.

We define the set of relaxation factors as

$$\bar{\mathcal{C}} := \left\{ c - c' : c, c' \in \mathcal{C} \text{ and } c \neq c' \right\},$$

i.e. set of differences of any two distinct elements in $\mathcal{C}$. We will choose the constant ν such that (experimentally) the probability for $c \leftarrow S_\xi$ to satisfy $\sqrt[2k]{\|c^{2k}\|_1} \leq \nu$ is at least 99%. In Figure 8 we show example parameters (ξ, ν, k) and the lower-bound on the size of $\mathcal{C}$.

$\hat{d}$	ξ	ν	k	$\lvert\mathcal{C}\rvert$
64	8	140	32	2^{129}
128	2	59	32	2^{147}

Fig. 8: Example parameters to instantiate $\mathcal{C}$ for a modulus $\hat{q}$ such that its smallest prime divisor p_1 is greater than 16.

For security of our protocols, we need the invertibility property of the set $\mathcal{C}$, i.e. the difference of any two distinct elements of $\mathcal{C}$ is invertible over $\hat{\mathcal{R}}_{\hat{q}}$. Hence, we apply [LNP22, Lemma 2.6] and thus we only need the condition $\xi < \hat{p}_1/2$ where $\hat{p}_1$ is the smallest prime dividing q. Secondly, to achieve negligible soundness error, we will need $|\mathcal{C}|$ to be exponentially large.

Now, we are ready to define the ABDLOP relaxed openings.

Definition 5.1. *A relaxed opening of the* ABDLOP *commitment* $(\mathbf{t}_A, \mathbf{t}_B)$ *w.r.t. a commitment key* crs *is a tuple* $(\boldsymbol{s}_1, \boldsymbol{m}, \boldsymbol{s}_2, c) \in \hat{\mathcal{R}}_{\hat{q}}^{m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{\ell} \times \hat{\mathcal{R}}_{\hat{q}}^{m_2} \times \bar{\mathcal{C}}$ *which satisfies:*

$$\mathsf{ABDLOP.Com}(\mathsf{crs}, \boldsymbol{s}_1, \boldsymbol{m}; \boldsymbol{s}_2) = (\mathbf{t}_A, \mathbf{t}_B)$$
$$\|c\boldsymbol{s}_1\| \leq \mathcal{B}_1 \quad \textit{and} \quad \|c\boldsymbol{s}_2\| \leq \mathcal{B}_2.$$

Security properties. As shown in [LNP22, Lemma 3.1], the ABDLOP commitment is binding with respect to relaxed openings under the Module-SIS assumption. As for hiding, it follows from the fact that under the Module-LWE assumption $\begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \boldsymbol{s}_2$ looks pseudorandom.

Lemma 5.2 (Binding). *Let $\xi < p_1/2$ where p_1 is the smallest prime dividing $\hat{q}$. Then, the* ABDLOP *commitment is computationally binding with respect to relaxed openings under the* $\mathsf{MSIS}_{n,m_1+m_2,\mathcal{B}}$ *assumption where* $\mathcal{B} := 4\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$.

[13] Message $\boldsymbol{m}$ does not need to be included in the opening since it can be deterministically computed from $\mathbf{t}_B$ and $\boldsymbol{s}_2$.
[14] Looking ahead, this set will be a challenge space and will play a big role in our protocols.

Lemma 5.3 (Hiding). *The* ABDLOP *commitment is computationally hiding under the* $\mathsf{MLWE}_{n+\ell,m_2,\chi}$ *assumption.*

Approximate Range Proofs. We use the results developed in [GHL21; LNP22] to upper-bound the operator norm of a matrix $\mathfrak{R} \leftarrow \mathsf{Bin}_\xi^{256\times m}$. Here, Bin denotes the binomial distribution with a positive integer parameter ξ, which is the distribution $\sum_{i=1}^{\xi}(a_i - b_i)$, where $a_i, b_i \leftarrow \{0,1\}$. The variance of this distribution is $\xi/2$ and it holds that $\mathsf{Bin}_{\xi_1} \pm \mathsf{Bin}_{\xi_2} = \mathsf{Bin}_{\xi_1+\xi_2}$.

The bound on the operator norm of $\mathfrak{R}$ is used for the *approximate range proof* part, where we want to show that $\vec{s}$ has relatively small coefficients w.r.t. to the proof system modulus $\hat{q}$. Namely, let us define functions $\omega_{\min}(\lambda)$ and $\omega_{\max}(\lambda)$ such that for any $\vec{w} \in \mathbb{Z}^m$ and any $\xi \in \mathbb{N}$:

$$\Pr_{\mathfrak{R}\leftarrow\mathsf{Bin}_\xi^{256\times m}}[\|\mathfrak{R}\vec{w}\|^2 > \xi \cdot \|\vec{w}\|^2 \cdot \omega_{\min}(\lambda)^2] \le 2^{-2\lambda},$$
$$\Pr_{\mathfrak{R}\leftarrow\mathsf{Bin}_\xi^{256\times m}}[\|\mathfrak{R}\vec{w}\|^2 < \xi \cdot \|\vec{w}\|^2 \cdot \omega_{\max}(\lambda)^2] \le 2^{-\lambda}.$$

It was shown in [GHL21] (under a few natural heuristic assumptions) that $\omega_{\min}(128) = \sqrt{13}$ and $\omega_{\max}(128) = \sqrt{337}$. In the soundness argument, we will use the following result from [LNP22, Lemma 2.9] in a slightly more general form (if one substitutes $\lambda = 128$ then it is identical to the aforementioned lemma):

Lemma 5.4. *Fix $m, P \in \mathbb{N}$ and a bound $b \le P/(8\sqrt{2}\cdot\omega_{\min}(\lambda)\cdot m)$, and let $\vec{w} \in [\pm P/2]^m$ with $\|\vec{w}\| \ge b$, and let $\vec{y}$ be an arbitrary vector in $[\pm P/2]^m$. Then*

$$\Pr_{R\leftarrow\mathsf{Bin}_1^{256\times m}}\left[\|R\vec{w} + \vec{y} \bmod P\| < \frac{b}{\sqrt{2}}\cdot\omega_{\min}(\lambda)\right] < 2^{-\lambda}.$$

5.2 The Protocol

We provide a brief summary of the Fiat-Shamir transformed [FS86] protocol from [LNP22] to prove (12). We assume that variables $\mathsf{n}, \mathsf{m}, \mathsf{t}, \ell_r$ are divisible by $\hat{d}$. First, note that by the sum-of-four-squares theorem, $\|\vec{s}\| \le \mathcal{B}_s$ is equivalent to the existence of four integers $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ such that $\mathcal{B}^2 - \|\vec{s}\|^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2$. Thus, we can define a vector $a := (a_1, a_2, a_3, a_4, 0, \ldots, 0) \in \mathbb{Z}_{\hat{q}}^{\hat{d}}$ and observe that:

$$\left\|\begin{bmatrix}\vec{s}\\ \vec{a}\end{bmatrix}\right\| = \mathcal{B}_s \quad \text{and} \quad \begin{bmatrix}P\ 0\end{bmatrix}\begin{bmatrix}\vec{s}\\ \vec{a}\end{bmatrix} = P\vec{s}.$$

Hence, we slightly abuse the notation and define $\mathsf{m} := \mathsf{m} + \hat{d}, \vec{s} := (\vec{s}, \vec{a}) \in \mathbb{Z}_{\hat{q}}^{\mathsf{m}}$ and $P := [P\ 0] \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{n}\times\mathsf{m}}$. Similarly, we proceed for the vector $\vec{r}$. Thus, we now need to prove that

$$\|\vec{s}\| = \mathcal{B}_s, \quad \|\vec{r}\| = \mathcal{B}_r, \quad \vec{u} \in \{0,1\}^{\mathsf{n}}, \quad P\vec{s} = B\vec{u} + C\begin{bmatrix}\vec{m}\\ \vec{r}\end{bmatrix}. \tag{14}$$

In this proof system, the common *random* string simply consists of uniformly random matrices which are only used for the ABDLOP commitment scheme [LNP22] :

$$\mathsf{crs} := (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}) \in \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{256/\hat{d}\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{\tau\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{m_2}. \tag{15}$$

We define the statement as the tuple:

$$\mathsf{x} := (P, C, \vec{m}, B, \mathsf{Bounds} := (\mathcal{B}_s, \mathcal{B}_r), \mathsf{aux}).$$

First round. Let $\mathbf{s} = \mathsf{Coeffs}^{-1}(\vec{s}) \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{m}/\hat{d}}$ and similarly for $\mathbf{r}, \mathbf{u}$. To begin with, we commit to the witness $(\mathbf{s}, \mathbf{r})$. To be consistent with notation from [LNP22] let $\mathbf{s}_1 := (\mathbf{s}, \mathbf{r}, \mathbf{u})$ and $m_1 = \mathsf{m} + \ell_r + \mathsf{t}$. We commit to $\mathbf{s}_1$ using the ABDLOP commitment (which in this case is just the Ajtai commitment), i.e. we sample the randomness vector $\mathbf{s}_2 \leftarrow \chi^{m_2}$, where χ is a probability distribution on ternary polynomials in $\hat{\mathcal{R}}_{\hat{q}}$, and compute:

$$\mathbf{t}_A := \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \quad \text{where} \quad (\mathbf{A}_1, \mathbf{A}_2) \in \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_2}.$$

Then, we sample short masking vectors $\mathbf{y}_i \leftarrow D_{\mathsf{s}_i}^{m_i}$ for $i = 1, 2$, and compute $\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$. Finally, we focus on the parts used to prove (14). Namely, we generate the polynomial vector $\mathbf{y}_3 \leftarrow D_{\mathsf{s}_3}^{256}$ and commit to it as follows

$$\mathbf{t}_y := \mathbf{B}_y\mathbf{s}_2 + \mathbf{y}_3 \quad \text{where} \quad \mathbf{B}_y \in \hat{\mathcal{R}}_{\hat{q}}^{\frac{256}{\hat{d}}\times m_2}.$$

Finally, we sample a polynomial vector $\mathbf{g} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$ and commit to it:

$$\mathbf{t}_g := \mathbf{B}_g\mathbf{s}_2 + \mathbf{g} \quad \text{where} \quad \mathbf{B}_g \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{\tau\times m_2}.$$

Hence, the first message and the corresponding challenge are

$$a_1 := (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}) \quad \text{and} \quad (\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1) \in \{0,1\}^{256\times m_1\hat{d}} \times \{0,1\}^{256\times m_1\hat{d}}.$$

Second round. Given the first challenge $(\mathfrak{R}_0, \mathfrak{R}_1)$, we compute the response

$$\vec{z}_3 := \vec{y}_3 + \mathfrak{R}\vec{s}_1 \in \mathbb{Z}^{256},$$

where $\vec{y}_3 := \mathsf{Coeffs}(\mathbf{y}_3), \vec{s}_1 := \mathsf{Coeffs}(\mathbf{s}_1)$, and $\mathfrak{R} := \mathfrak{R}_0 - \mathfrak{R}_1$, and apply rejection sampling on $\vec{z}_3$. The second message and the corresponding challenge are:

$$a_2 := \vec{z}_3 \quad \text{and} \quad (\gamma_{i,j})_{i\in[\tau],j\in[256+\mathsf{n}+3]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2) \in \mathbb{Z}_{\hat{q}}^{\tau\times(256+\mathsf{n}+3)}.$$

Third round. The prover's goal now is to prove the following relations over $\mathbb{Z}_{\hat{q}}$:

$$\begin{cases} \vec{z}_3 = \vec{y}_3 + \mathfrak{R}\vec{s}_1 \\ P\vec{s} = B\vec{u} + C\begin{bmatrix}\vec{m}\\ \vec{r}\end{bmatrix} \\ \langle\vec{s}, \vec{s}\rangle = \mathcal{B}_s^2 \\ \langle\vec{r}, \vec{r}\rangle = \mathcal{B}_r^2 \\ \langle\vec{u}, \vec{u} - \vec{1}\rangle = 0 \end{cases} \tag{16}$$

where the terms in blue are secret and committed. We will use the following observation from [LNP22, Lemma 2.4]: for any integer vectors $\vec{x}_0, \vec{x}_1 \in \mathbb{Z}_{\hat{q}}^n$, the constant coefficient of

$$\mathsf{Coeffs}^{-1}\left(\vec{x}_0\right)^T \sigma\left(\mathsf{Coeffs}^{-1}(\vec{x}_1)\right) \in \hat{\mathcal{R}}_{\hat{q}}$$

is equal to $\langle\vec{x}_0, \vec{x}_1\rangle \in \mathbb{Z}_{\hat{q}}$. This implies that proving the first equation is equivalent to proving that for $i \in [256]$, the constant coefficient of

$$\sigma(\mathbf{r}_i)^T\mathbf{s}_1 + \sigma(\mathbf{e}_i)^T\mathbf{y}_3 - z_{3,i} \text{ is equal to zero,}$$

where $\mathbf{r}_i$ is the polynomial vector such that its coefficient vector is the i-th row of $\mathfrak{R}$, and $\mathbf{e}_i$ is the polynomial vector so that $\mathsf{Coeffs}(\mathbf{e}_i)$ is a unit vector[15] with its i-th coefficient being 1. Similarly, the second equation is equivalent to: for all $i \in [\mathsf{n}]$, the constant coefficient of:

$$\sigma(\mathbf{p}_i)^T\mathbf{s} - \sigma(\boldsymbol{\beta}_i)^T\mathbf{u} - m_{C,i} - \sigma(\mathbf{c}_{r,i})^T\mathbf{r} \text{ is equal to zero,}$$

where:

[15] Concretely, it is a binary vector with exactly one 1 entry.

– $\mathbf{p}_i$ is the polynomial vector with its coefficients being the i-th row of P,
– $\boldsymbol{\beta}_i$ is the polynomial vector with its coefficients being the i-th row of B,
– we write $C = [C_m\ C_r]$ such that $C \begin{bmatrix} \vec{m} \\ \vec{r} \end{bmatrix} = C_m\vec{m} + C_r\vec{r}$, and define $\vec{m}_C := C_m\vec{m}$,
– $\mathbf{c}_{r,i}$ is the polynomial vector with its coefficients being the i-th row of C_r.

Clearly, the last three equations are equivalent to

$$\sigma(\mathbf{s})^T\mathbf{s} - \mathcal{B}_s, \quad \sigma(\mathbf{r})^T\mathbf{r} - \mathcal{B}_r, \quad \sigma\left(\mathbf{u} - \mathsf{Coeffs}^{-1}(1^{\mathsf{t}})\right)^T\mathbf{u}$$

having the constant coefficient equal to zero. Hence, we compute for $i \in [\tau]$:

$$\begin{aligned} h_i := g_i &+ \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T\mathbf{s}_1 + \sigma(\mathbf{e}_j)^T\mathbf{y}_3 - z_j\right) \\ &+ \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T\mathbf{s} - \sigma(\boldsymbol{\beta}_j)^T\mathbf{u} - m_{C,j} - \sigma(\mathbf{c}_{r,j})^T\mathbf{r}\right) \\ &+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\mathbf{s})^T\mathbf{s} - \mathcal{B}_s\right) + \gamma_{i,256+\mathsf{n}+2} \cdot \left(\sigma(\mathbf{r})^T\mathbf{r} - \mathcal{B}_r\right) \\ &+ \gamma_{i,256+\mathsf{n}+3} \cdot \sigma\left(\mathbf{u} - \mathsf{Coeffs}^{-1}(1^{\mathsf{t}})\right)^T\mathbf{u} \in \hat{\mathcal{R}}_{\hat{q}}. \end{aligned} \tag{17}$$

Observe that by definition of $\mathbf{g}$, the constant coefficients of $h_1, \ldots, h_\tau$ are all zeroes. The third message and the corresponding challenge are:

$$a_3 := (h_1, \ldots, h_\tau) \quad \text{and} \quad \boldsymbol{\mu} = (\mu_i)_{i\in[\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3) \in \hat{\mathcal{R}}_{\hat{q}}^{\tau}.$$

Fourth round. The goal of the prover is now to prove the τ quadratic equations (with automorphisms) over $\mathcal{R}_{\hat{q}}$ from (17). We now simply linear-combine them with the challenges μ_i, i.e. we prove a single equation:

$$\begin{aligned} 0 = \sum_{i=1}^{\tau} \mu_i \Bigg(&\sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T\mathbf{s}_1 + \sigma(\mathbf{e}_j)^T\mathbf{y}_3 - z_{3,j}\right) \\ &+ \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T\mathbf{s} - \sigma(\boldsymbol{\beta}_j)^T\mathbf{u} - m_{C,j} - \sigma(\mathbf{c}_{r,j})^T\mathbf{r}\right) \\ &+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\mathbf{s})^T\mathbf{s} - \mathcal{B}_s\right) + \gamma_{i,256+\mathsf{n}+2} \cdot \left(\sigma(\mathbf{r})^T\mathbf{r} - \mathcal{B}_r\right) \\ &+ \gamma_{i,256+\mathsf{n}+3} \cdot \sigma\left(\mathbf{u} - \mathsf{Coeffs}^{-1}(1^{\mathsf{t}})\right)^T\mathbf{u} + g_i - h_i \Bigg). \end{aligned} \tag{18}$$

Let us define

$$\mathbf{B} := \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix}, \quad \mathbf{t}_B := \begin{bmatrix} \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix}, \quad \mathbf{m} := \begin{bmatrix} \mathbf{y}_3 \\ \mathbf{g} \end{bmatrix} \in \hat{\mathcal{R}}_{\hat{q}}^{256/\hat{d}+\tau} \quad \text{and} \quad \hat{\mathbf{s}} := \begin{bmatrix} \mathbf{s}_1 \\ \sigma(\mathbf{s}_1) \\ \mathbf{m} \\ \sigma(\mathbf{m}) \end{bmatrix}. \tag{19}$$

Recall that $\mathbf{s}_1 = (\mathbf{s}, \mathbf{r}, \mathbf{u})$ and let us write $\mathfrak{r}_j := (\mathfrak{r}_{s,j}, \mathfrak{r}_{r,j}, \mathfrak{r}_{u,j})$ such that

$$\sigma(\mathbf{r}_j)^T\mathbf{s}_1 = \sigma(\mathfrak{r}_{s,j})^T\mathbf{s} + \sigma(\mathfrak{r}_{r,j})^T\mathbf{r} + \sigma(\mathfrak{r}_{u,j})^T\mathbf{u}.$$

Then, the quadratic equation above can be written equivalently as

$$\hat{\mathbf{s}}^T \mathbf{D}_2 \hat{\mathbf{s}} + \mathbf{d}_1^T \hat{\mathbf{s}} + d_0 = 0$$

where

$$\mathbf{D}_2 := \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \sum_{i=1}^{\tau} \mu_i \gamma_{i,256+\mathsf{n}+1} \cdot \mathbf{I}_{\mathsf{m}/\hat{d}} & \sum_{i=1}^{\tau} \mu_i \gamma_{i,256+\mathsf{n}+2} \cdot \mathbf{I}_{\ell_r/\hat{d}} & \sum_{i=1}^{\tau} \mu_i \gamma_{i,256+\mathsf{n}+3} \cdot \mathbf{I}_{\mathsf{n}/\hat{d}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{d}_1 := \begin{bmatrix} \sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} \gamma_{i,j} \sigma(\mathbf{r}_{s,j}) + \sum_{j'=1}^{\mathsf{n}} \gamma_{i,256+j'} \sigma(\mathbf{p}_{j'}) \right) \\ \sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} \gamma_{i,j} \sigma(\mathbf{r}_{r,j}) - \sum_{j'=1}^{\mathsf{n}} \gamma_{i,256+j'} \sigma(\mathbf{c}_{r,j'}) \right) \\ \sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} \gamma_{i,j} \sigma(\mathbf{r}_{u,j}) - \sum_{j'=1}^{\mathsf{n}} \gamma_{i,256+j'} \sigma(\boldsymbol{\beta}_{j'}) - \gamma_{i,256+\mathsf{n}+3} \mathsf{Coeffs}^{-1}(1^{\mathsf{t}}) \right) \\ \mathbf{0} \\ \sum_{i=1}^{\tau} \sum_{j=1}^{256} \mu_i \gamma_{i,j} \sigma(\mathbf{e}_j) \\ \boldsymbol{\mu} \\ \mathbf{0} \end{bmatrix} \tag{20}$$

$$d_0 := -\sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} z_{3,j} + \sum_{j'=1}^{\mathsf{n}} m_{C,j'} + \gamma_{i,256+\mathsf{n}+1} \mathcal{B}_s + \gamma_{i,256+\mathsf{n}+2} \mathcal{B}_r + h_i \right).$$

Eventually, we run the sub-protocol for proving a single quadratic equation with automorphisms. First, we calculate the garbage term $f_1 = \hat{\mathbf{s}}^T \mathbf{R}_2 \mathbf{y} + \mathbf{y}^T \mathbf{R}_2 \hat{\mathbf{s}} + \mathbf{r}_1^T \mathbf{y}$, where $\mathbf{y}$ is defined as

$$\mathbf{y} := \begin{bmatrix} \mathbf{y}_1 \\ \sigma(\mathbf{y}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ -\sigma(\mathbf{B}\mathbf{y}_2) \end{bmatrix} \in \hat{\mathcal{R}}_{\hat{q}}^{2(m_1+256/d+\tau)}, \tag{21}$$

and the commitment $t = \mathbf{b}^T \mathbf{s}_2 + f_1$ to f_1. Then, we set $f_0 = \mathbf{y}^T \mathbf{R}_2 \mathbf{y} + \mathbf{b}^T \mathbf{y}_2$. Hence, the fourth message and the corresponding challenge are (we defined $\mathcal{C}$ in Section 5.1):

$$a_4 := (t, f_0) \quad \text{and} \quad c := \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4) \in \mathcal{C}.$$

Final round. Given a challenge c, we compute $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ for $i = 1, 2$ and apply rejection sampling on $\mathbf{z}_1, \mathbf{z}_2$. So, the prover's last message is $a_5 := (\mathbf{z}_1, \mathbf{z}_2)$ and thus the proof consists of

$$\pi := (a_1, a_2, a_3, a_4, a_5).$$

Verification algorithm. The verifier is given a proof π and recomputes the corresponding challenges as well as $\mathbf{D}_2, \mathbf{d}_1, d_0$. Let us define

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \sigma(\mathbf{z}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \end{bmatrix}. \tag{22}$$

Finally, the verifier outputs 1 if *all* the following relations hold:

$$\begin{cases} \|\mathbf{z}_1\| \overset{?}{\leq} \mathcal{B}_1, \|\mathbf{z}_2\| \overset{?}{\leq} \mathcal{B}_2, \|\vec{z}_3\| \overset{?}{\leq} \mathcal{B}_3, \\ \tilde{h}_i \overset{?}{=} 0 \text{ for } i \in [\tau], \\ \mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}_2 \mathbf{z}_2 \overset{?}{=} \mathbf{w} + c\mathbf{t}_A, \\ \mathbf{z}^T \mathbf{D}_2 \mathbf{z} + c\mathbf{d}_1^T \mathbf{z} + c^2 d_0 - (ct - \mathbf{b}^T \mathbf{z}_2) \overset{?}{=} f_0 \end{cases}$$

and 0 otherwise.

For completeness, we provide a description of the NIZK proof system $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}} = (\mathsf{Prove}^{\mathsf{H}}_{\mathsf{ISIS}}, \mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}})$ for proving (12) in Figures 9 and 10. Similarly as in many prior works, e.g. [Lyu12], in practice we do not include $\mathbf{w}, f_0$ in the proof and send c instead, since these components can be computed deterministically given other parts of π and c. However, we keep the standard Fiat-Shamir transformation for simpler security analysis.

5.3 Security Analysis

In this subsection, we prove the key properties of $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$, i.e. correctness and zero-knowledge. As for knowledge soundness, we will prove it directly when arguing unforgeability of our anonymous credentials in Section 8.

5.3.1 Correctness

Lemma 5.5 (Correctness). *Let $\alpha_1, \alpha_2, \alpha_3, \bar{\alpha}_1, \bar{\alpha}_2 \in O(\sqrt{\lambda})$ and $\mathsf{N}, \hat{d} \in O(\lambda)$. Further, suppose χ is a probability distribution over S_1. Recall $\nu, \omega_{\max}$ from Section 5.1, and define the following variables:*

$$\mathfrak{s}_1 := \alpha_1 \nu \sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}}, \quad \mathfrak{s}_2 := \alpha_2 \nu \sqrt{m_2 \hat{d}}, \quad \mathfrak{s}_3 := \alpha_3 \omega_{\max}(\lambda) \sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}},$$
$$\mathcal{B}_1 := \mathfrak{s}_1 \sqrt{2 m_1 \hat{d}}, \quad \mathcal{B}_2 := \mathfrak{s}_2 \sqrt{2 m_2 \hat{d}}, \quad \mathcal{B}_3 := 1.7 \mathfrak{s}_3 \sqrt{256}$$

and the corresponding rejection rates:

$$M_i := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\alpha_i} + \frac{1}{2\alpha_i^2} \right) \text{ for } i \in \{1,2,3\}.$$

Then, $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}} := (\mathsf{Prove}^{\mathsf{H}}_{\mathsf{ISIS}}, \mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}})$ is correct.

Proof. First, one observes that both algorithms, and especially $\mathsf{Prove}^{\mathsf{H}}_{\mathsf{ISIS}}$, are *strict* polynomial time since we bounded the number of restarts by $O(\lambda)$. Next, we argue that the prover algorithm outputs $\perp$ with negligible probability. First, using Lemma 2.4 and the following inequalities (which hold with an overwhelming probability):

$$\|c\mathbf{s}_1\| \le \nu \sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}}, \quad \|c\mathbf{s}_2\| \le \nu \sqrt{m_2 \hat{d}}, \quad \|\mathfrak{R}\vec{s}_1\| \le \omega_{\max}(\lambda) \cdot \sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}}.$$

Hence, we deduce that in a single run the probability that $b_1 b_2 b_3 = 1$ is at least $(M_1 M_2 M_3)^{-1} - \mathsf{negl}(\lambda)$ by Lemma 2.4, and thus the probability that $\mathsf{Prove}^{\mathsf{H}}_{\mathsf{ISIS}}$ outputs $\perp$, i.e. makes exactly N unsuccessful attempts, is at most

$$\left(1 - \frac{1}{M_1 M_2 M_3} + \mathsf{negl}(\lambda)\right)^{\mathsf{N}} = \mathsf{negl}(\lambda)$$

since $M_1, M_2, M_3 \in O(1)$ and $\mathsf{N} = O(\lambda)$.

In terms of verification equations, by Lemma 2.2 for $t = \sqrt{2}$ and the union bound, the probability that $\|\mathbf{z}_1\| \le \mathfrak{s}_1 \sqrt{2 m_1 \hat{d}}$ and $\|\mathbf{z}_2\| \le \mathfrak{s}_2 \sqrt{2 m_2 \hat{d}}$ is overwhelming. We argue similarly for $\|\vec{z}_3\| > 1.7 \mathfrak{s}_3 \sqrt{256}$ where we apply Lemma 2.2 for $t = 1.7$ instead [16]. The rest of the verification checks hold by careful inspection of the protocol (we refer to [LNP22] for more details). □

[16] The reason is to ensure $\approx 2^{-128}$ correctness error when setting the parameters.

$\underline{\mathsf{Prove}_{\mathsf{ISIS}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})}$

Input: $\mathsf{crs} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b})$, $\mathsf{x} = (P, C, \vec{m}, B, \mathsf{Bounds}, \mathsf{aux})$, $\mathsf{w} = (\vec{s}, \vec{r}, \vec{u})$

Output: proof π

$\mathsf{rst} = 0$ ▷ boolean which indicates if a restart is necessary

$\mathsf{idx} = 0$ ▷ keeps track how many restarts occured

$\mathbf{s}_1 = (\mathbf{s}, \mathbf{r}, \mathbf{u})$ ▷ $\mathbf{s}, \mathbf{r}, \mathbf{u}$ are the polynomial vectors with coeff. $\vec{s}, \vec{r}, \vec{u}$

while $\mathsf{rst} = 0 \wedge \mathsf{idx} < \mathsf{N}$ **do**

  $\mathsf{idx} = \mathsf{idx} + 1$

  $\mathbf{s}_2 \leftarrow \chi^{m_2}$ ▷ generate commitment randomness

  $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$ ▷ Ajtai commitment to $\mathbf{s}_1$

  $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1\hat{d}}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2\hat{d}}, \mathbf{y}_3 \leftarrow D_{\mathfrak{s}_3}^{256}$ ▷ sample masking vectors

  $\mathbf{g} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$ ▷ sample random polynomials with const. coeff. zero

  $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$ ▷ used to prove knowledge of a commitment opening

  $\mathbf{t}_y = \mathbf{B}_y\mathbf{s}_2 + \mathbf{y}_3$ ▷ commitment to $\mathbf{y}_3$

  $\mathbf{t}_g = \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}$ ▷ commitment to $\mathbf{g}$

  $a_1 = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w})$ ▷ first message

  $(\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1) \in \{0,1\}^{256\times m_1\hat{d}} \times \{0,1\}^{256\times m_1\hat{d}}$ ▷ first challenge

  $\mathfrak{R} = \mathfrak{R}_0 - \mathfrak{R}_1$

  $\vec{s}_1 = \mathsf{Coeffs}(\mathbf{s}_1)$

  $\vec{y}_3 = \mathsf{Coeffs}(\mathbf{y}_3)$

  $\vec{z}_3 = \vec{y}_3 + \mathfrak{R}\vec{s}_1 \in \mathbb{Z}^{256}$ ▷ masked opening of $\mathfrak{R}\vec{s}_1$

  $b_3 \leftarrow \mathsf{Rej}\left(\vec{z}_3, \mathfrak{R}\vec{s}_1, \mathfrak{s}_3, M_3\right)$

  $a_2 = \vec{z}_3$ ▷ second message

  $(\gamma_{i,j})_{i\in[\tau],j\in[256+\mathsf{n}+3]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2) \in \mathbb{Z}_{\hat{q}}^{\tau\times(256+\mathsf{n}+3)}$ ▷ second challenge

  for $i \in [\tau]$ **do**

    compute h_i as in (17)

  $a_3 = \mathbf{h} = (h_1, \ldots, h_\tau)$ ▷ third message

  $(\mu_i)_{i\in[\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3) \in \hat{\mathcal{R}}_{\hat{q}}^{\tau}$ ▷ third challenge

  compute $\mathbf{B}, \mathbf{t}_B, \mathbf{m}, \hat{\mathbf{s}}, \mathbf{y}$ as in (19) and (21)

  compute $\mathbf{D}_2, \mathbf{d}_1, d_0$ as in (20)

  $f_1 = \mathbf{s}^T\mathbf{R}_2\mathbf{y} + \mathbf{y}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{y}$

  $f_0 = \mathbf{y}^T\mathbf{R}_2\mathbf{y} + \mathbf{b}^T\mathbf{y}_2$

  $t = \mathbf{b}^T\mathbf{s}_2 + f_1$ ▷ commitment to f_1

  $a_4 = (t, f_0)$ ▷ fourth message

  $c = \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4) \in \mathcal{C}$ ▷ fourth challenge

  for $i \in \{1, 2\}$ **do**

    $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ ▷ maked opening of $c\mathbf{s}_i$

    $b_i \leftarrow \mathsf{Rej}\left(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i, M_i\right)$ ▷ rejection sampling

  $\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2)$ ▷ candidate proof π

  $\mathsf{rst} = b_1b_2b_3$ ▷ if all rejection steps passed then $\mathsf{rst} = 1$

if $\mathsf{rst} = 1$ **then return** π

else return $\perp$

Fig. 9: $\mathsf{Prove}_{\mathsf{ISIS}}^{\mathsf{H}}$ algorithm.

$\mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}}(\mathsf{crs}, \mathsf{x}, \pi)$
Input: $\mathsf{crs} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b})$, $\mathsf{x} = (P, C, \vec{m}, B, \mathsf{Bounds}, \mathsf{aux})$, π
Output: bit b

1: parse $\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2)$
2: $a_1 = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w})$
3: $a_2 = \vec{z}_3$
4: $a_3 = \mathbf{h}$
5: $a_4 = (t, f_0)$
6: $(\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1)$ ▷ recomputing the challenges
7: $(\gamma_{i,j})_{i\in[\tau], j\in[256+\mathsf{n}+3]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2)$
8: $(\mu_i)_{i\in[\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3)$
9: $c = \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4)$
10: $\mathbf{B} = \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix}, \mathbf{t}_B = \begin{bmatrix} \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix}$
11: $\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \sigma(\mathbf{z}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \end{bmatrix}$ ▷ grouping the masked openings to $c\mathbf{s}_1$ and $c\mathbf{m}$
12: compute $\mathbf{D}_2, \mathbf{d}_1, d_0$ as in (20)
13: **if** one of the conditions below does not hold, then **return** 0:
    1. $\|\mathbf{z}_1\| \overset{?}{\leq} \mathcal{B}_1, \|\mathbf{z}_2\| \overset{?}{\leq} \mathcal{B}_2, \|\vec{z}_3\| \overset{?}{\leq} \mathcal{B}_3$
    2. $\tilde{h}_i \overset{?}{=} 0$ for $i \in [\tau]$
    3. $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 \overset{?}{=} \mathbf{w} + c\mathbf{t}_A$
    4. $\mathbf{z}^T\mathbf{D}_2\mathbf{z} + c\mathbf{d}_1^T\mathbf{z} + c^2 d_0 - (ct - \mathbf{b}^T\mathbf{z}_2) \overset{?}{=} f_0$
14: **else return** 1 ▷ all the verification checks passed

Fig. 10: $\mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}}$ algorithm.

5.3.2 Zero-Knowledge

Lemma 5.6 (Zero-Knowledge). *Let $m_2 \geq 256/\hat{d} + \tau + 1$, $\alpha_1, \alpha_2, \alpha_3 \in O(\sqrt{\lambda})$ and $\mathsf{N}, d \in O(\lambda)$. Define the following variables:*

$$\mathfrak{s}_1 := \alpha_1\nu\sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}}, \quad \mathfrak{s}_2 := \alpha_2\nu\sqrt{m_2\hat{d}}, \quad \mathfrak{s}_3 := \alpha_3\omega_{\max}(\lambda)\sqrt{\mathcal{B}_s^2 + \mathcal{B}_r^2 + \mathsf{t}},$$
$$\mathcal{B}_1 := \mathfrak{s}_1\sqrt{2m_1\hat{d}}, \quad \mathcal{B}_2 := \mathfrak{s}_2\sqrt{2m_2\hat{d}}, \quad \mathcal{B}_3 := 1.7\mathfrak{s}_3\sqrt{256}$$

and the corresponding rejection rates:

$$M_i := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\alpha_i} + \frac{1}{2\alpha_i^2}\right) \text{ for } i \in \{1, 2, 3\}.$$

Then, under the $\mathsf{MLWE}_{n+256/\hat{d}+\tau+1, m_2, \chi, \hat{q}}$ assumption $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ is zero-knowledge.

Proof. We prove the statement using a hybrid argument. Let $\mathcal{A}$ be a PPT algorithm, which makes at most Q_0 queries to H and Q_1 queries to $\mathsf{Prove}_{\mathsf{ISIS}}$, outputs 1 with probability ϵ. In each i-th game, we will introduce PPT algorithms $\mathcal{S}_0, \mathcal{S}_1$, which have a shared state, and denote ε_i to be the probability that $1 \leftarrow \mathcal{A}^{\mathcal{S}_0, \mathcal{S}_1}(\mathsf{crs})$.

Game_1: Here, $\mathcal{S}_0$ simulates the random oracle queries using lazy sampling and $\mathcal{S}_1$ runs $\mathsf{Prove}_{\mathsf{ISIS}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$ truthfully. By construction, $\epsilon_1 = \epsilon$.

Game_2: When $\mathcal{S}_1$ is queried, before executing Line 6 it first samples $c \leftarrow \mathcal{C}$. Then in Line 32, instead of calling

the random oracle $\mathcal{S}_0$, it uses the challenge c generated at the very beginning. If $(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4)$ was already queried earlier to $\mathcal{S}_0$, then $\mathcal{S}_1$ aborts. Otherwise, it continues with the $\mathsf{Prove}_{\mathsf{ISIS}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$ algorithm.

We argue the indistinguishability following the proof of [Lyu12, Lemma 5.3]. Namely, the only difference between Game_1 and Game_2 is that we directly program c in Line 32 without checking whether $\mathsf{a} := (4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4)$ was already set. Since $\mathcal{A}$ calls $\mathcal{S}_0$ at most Q_0 times and $\mathcal{S}_1$ at most Q_1 times, at most $Q_0 + \mathsf{N}Q_1$ of such tuples a will ever be set. Recall that a, and especially a_1, contains $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$. Therefore, at most $Q_0 + \mathsf{N}Q_1$ values of $\mathbf{w}$ will be set. Also, with an overwhelming probability over the standard parameter choices [EZS+19, Appendix C], matrix $\mathbf{A}_2$ can be written in the Hermite Normal Form as $\mathbf{A}_2 = \hat{\mathbf{A}}_2^{-1}[\mathbf{A}'_2 \ \mathbf{I}_n]$ and thus

$$\mathbf{y}_{2,2} = \hat{\mathbf{A}}_2(\mathbf{w} - \mathbf{A}_1\mathbf{y}_1) - \mathbf{A}'_2\mathbf{y}_{2,1}.$$

where $\mathbf{y}_2 := (\mathbf{y}_{2,1}, \mathbf{y}_{2,2}) \in \hat{\mathcal{R}}_{\hat{q}}^{m_2-(n+256/\hat{d}+\tau+1)} \times \hat{\mathcal{R}}_{\hat{q}}^{n+256/\hat{d}+\tau+1}$. Hence, by [Lyu12, Lemma 4.4], for any fixed $\mathbf{w} \in \hat{\mathcal{R}}_{\hat{q}}^n$ we have

$$\begin{aligned}\Pr\left[\mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 = \mathbf{w} : \mathbf{y}_1 \leftarrow D_{\mathsf{s}_2}^{m_1\hat{d}}, \mathbf{y}_2 \leftarrow D_{\mathsf{s}_2}^{m_2\hat{d}}\right] &\leq \max_{\mathbf{w}'_1 \in \hat{\mathcal{R}}_{\hat{q}}^n} \Pr\left[\mathbf{y}_{2,2} = \mathbf{w}'_1 : \mathbf{y}_{2,2} \leftarrow D_{\mathsf{s}_2}^{n\hat{d}}\right] \\ &\leq 2^{-n\hat{d}}.\end{aligned}$$

So, if $\mathcal{S}_1$ is accessed Q_1 times and the probability of getting a collision each time is at most $(Q_0 + \mathsf{N}Q_1)2^{-n\hat{d}}$, then the probability that a collision occurs after Q_1 queries is at most $\mathsf{N}Q_1(Q_0 + \mathsf{N}Q_1)2^{-n\hat{d}}$. Since $\mathsf{N}, d \in O(\lambda)$ and Q_0, Q_1 are polynomials in λ, we get

$$|\epsilon_2 - \epsilon_1| \leq \mathsf{N}Q_1(Q_0 + \mathsf{N}Q_1)2^{-n\hat{d}} + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda).$$

Game_3: Here, $\mathcal{S}_1$ directly computes $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ for $i = 1, 2$ and $\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$. Further, $\mathcal{S}_1$ later calculates $f_0 := \mathbf{z}^T\mathbf{D}_2\mathbf{z} + c\mathbf{d}_1^T\mathbf{z} + c^2d_0 - \mathsf{f}$, where $\mathsf{f} := ct - \mathbf{b}^T\mathbf{z}_2$, instead. By the verification equations and simple rearrangement argument, we have $\epsilon_3 = \epsilon_2$.

Game_5: Now, $\mathcal{S}_1$ directly samples each $\mathbf{z}_2 \leftarrow D_{\mathsf{s}_2}^{m_2\hat{d}}$ and sets $b_2 = 1$ with probability $1/M_2$. By Lemma 2.4, we have $|\epsilon_5 - \epsilon_4| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

Game_6: This time, $\mathcal{S}_1$ samples uniformly random $\boldsymbol{\nu} \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{n+256/\hat{d}+\tau+1}$ and computes the commitment the following way:

$$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_y \\ \mathbf{t}_g \\ t\end{bmatrix} := \boldsymbol{\nu} + \begin{bmatrix}\mathbf{A}\mathbf{s}_1 \\ \mathbf{y}_3 \\ \mathbf{g} \\ f_1\end{bmatrix}.$$

One can naturally construct a PPT adversary $\mathcal{A}^{\mathsf{MLWE}}$ which breaks Module-LWE with probability $|\epsilon_5 - \epsilon_4|/Q_1$. Hence,

$$|\epsilon_6 - \epsilon_5| \leq \mathsf{N} \cdot Q_1 \cdot \mathsf{Adv}^{\mathsf{MLWE}}_{n+256/\hat{d}+\tau+1, m_2, \chi}(\mathcal{A}^{\mathsf{MLWE}}) = \mathsf{negl}(\lambda).$$

Game_7: Here, $\mathcal{S}_1$ samples directly $(\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, t) \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{n+256/\hat{d}+\tau+1}$. Clearly, we have $\epsilon_7 = \epsilon_6$.

Game_8: Now, $\mathcal{S}_1$ generates $\mathbf{z}_1 \leftarrow D_{\mathsf{s}_1}^{m_1\hat{d}}$ and sets $b_1 = 1$ with probability $1/M_1$. By Lemma 2.4, we have $|\epsilon_8 - \epsilon_7| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

Game_9: Here, $\mathcal{S}_1$ picks $\mathbf{h} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^\tau$ directly. Since earlier $\mathbf{g}$ was chosen uniformly at random from $\{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^\tau$, we get $\epsilon_9 = \epsilon_8$.

Game_{10}: This time $\mathcal{S}_1$ samples $\vec{z}_3 \leftarrow D^{256}_{\mathsf{s}_3}$ and sets $b_3 = 1$ with probability $1/M_3$. By Lemma 2.4, $|\epsilon_{10} - \epsilon_9| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

Now, note that in the last security game $\mathcal{S}_1$ does not use the witness w (recall that $\mathcal{S}_0$ only simulates the random oracle H using lazy sampling). Hence, by the hybrid argument, $|\epsilon_{10} - \epsilon| = \mathsf{negl}(\lambda)$ which concludes the proof. □

6 Efficient Multi-Proof Extractable Non-Interactive Zero-Knowledge Proofs

This section focuses on constructing multi-proof extractable NIZK for proving knowledge of a short vector $\vec{s} \in \hat{\mathcal{R}}_{\hat{q}}^{m}$ such that $\|\vec{s}\| \leq \mathcal{B}$ and $P\vec{s} = \vec{u}$ over $\mathbb{Z}_{\hat{q}}$ where $\mathbf{P} \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{n}\times\mathsf{m}}$ and $\vec{u} \in \mathbb{Z}_{\hat{q}}^{\mathsf{n}}$ are public. We define the corresponding relation

$$R^{\mathsf{Com}} := \left\{ \mathsf{x} := (P, \vec{u}), \mathsf{w} := \vec{s} \middle| P \in \mathbb{Z}_{\hat{q}}^{\mathsf{n}\times\mathsf{m}}, \vec{u} \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{n}}, \vec{s} \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{m}} : P\vec{s} = \vec{u} \wedge \|\vec{s}\| \leq \mathcal{B} \right\}. \tag{23}$$

Throughout the section, we follow the footsteps of del Pino and Katsumata [PK22][Section 4.3] who added multi-proof extractability to the exact proof system by Bootle et al. [BLS19]. Namely, we adapt the recently proposed lattice-based framework for proving exact statements by Lyubashevsky et al. [LNP22] to prove relation R^{Com}. However, as shown in [LNP22, Appendix B], the protocol only satisfies standard knowledge soundness property. Hence, we further apply the Katsumata transform [Kat21] to achieve multi-proof extractability [17]

Conceptually, the protocol will be almost identical to the in Section 5. Hence, we use the same notation as in Section 5.1. As before, we assume that variables n, m are divisible by $\hat{d}$. By the sum-of-four-squares theorem, $\|\vec{s}\| \leq \mathcal{B}_s$ is equivalent to the existence of four integers $a_1, a_2, a_3, a_4 \in \mathbb{Z}$ such that $\mathcal{B}^2 - \|\vec{s}\|^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2$. Thus, we can define a vector $a := (a_1, a_2, a_3, a_4, 0, \ldots, 0) \in \mathbb{Z}_{\hat{q}}^{\hat{d}}$ and observe that:

$$\left\| \begin{bmatrix} \vec{s} \\ \vec{a} \end{bmatrix} \right\| = \mathcal{B}_s \quad \text{and} \quad \begin{bmatrix} P & 0 \end{bmatrix} \begin{bmatrix} \vec{s} \\ \vec{a} \end{bmatrix} = P\vec{s}.$$

Hence, we slightly abuse the notation and define $\mathsf{m} := \mathsf{m} + 1, \vec{s} := (\vec{s}, \vec{a}) \in \mathbb{Z}_{\hat{q}}^{\mathsf{m}}$ and $P := [P\ 0] \in \hat{\mathcal{R}}_{\hat{q}}^{\mathsf{n}\times\mathsf{m}}$. Therefore,

$$\|\vec{s}\| = \mathcal{B} \quad \text{and} \quad P\vec{s} = \vec{u}. \tag{24}$$

In this proof system, the common *random* string consists of uniformly random matrices which are only used for the ABDLOP commitment scheme:

$$\mathsf{crs} := (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}) \in \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{256/\hat{d}\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{\tau\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{m_2}. \tag{25}$$

We define the statement as the tuple $\mathsf{x} := (P, \vec{u}, \mathcal{B})$.

First round. Let $\mathbf{s} = \mathsf{Coeffs}^{-1}(\vec{s})$ and similarly for $\mathbf{u}$. We start by committing to the witness $\mathbf{s}_1 := \mathbf{s}$ and let $m_1 = \mathsf{m}$. We commit to $\mathbf{s}_1$ using the ABDLOP commitment, i.e. we sample the randomness vector $\mathbf{s}_2 \leftarrow \chi^{m_2}$, where χ is a probability distribution on ternary polynomials in $\hat{\mathcal{R}}_{\hat{q}}$, and compute:

$$\mathbf{t}_A := \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \quad \text{where} \quad (\mathbf{A}_1, \mathbf{A}_2) \in \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_2}.$$

Then, we sample short masking vectors $\mathbf{y}_i \leftarrow D^{m_i}_{\mathsf{s}_i}$ for $i = 1, 2$, and compute $\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$. Further, we generate the polynomial vector $\mathbf{y}_3 \leftarrow D^{256}_{\mathsf{s}_3}$ and commit to it as follows

$$\mathbf{t}_y := \mathbf{B}_y\mathbf{s}_2 + \mathbf{y}_3 \quad \text{where} \quad \mathbf{B}_y \in \hat{\mathcal{R}}_{\hat{q}}^{\frac{256}{\hat{d}}\times m_2}.$$

[17] We note that in the random oracle model one could apply the standard "encryption-to-the-sky" paradigm to obtain straight-line extractability (e.g. [AKSY22; BLNS23]). We provide the Katsumata transform instead since it has more potential to be further used to prove QROM security, as seen in [PK22]. We leave the quantum security analysis of the protocol, and of our constructions as future work.

Finally, we sample a polynomial vector $\mathbf{g} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$ and commit to it:

$$\mathbf{t}_g := \mathbf{B}_g \mathbf{s}_2 + \mathbf{g} \quad \text{where} \quad \mathbf{B}_g \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{\tau \times m_2}.$$

Hence, the first message and the corresponding challenge are

$$a_1 := (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}) \quad \text{and} \quad (\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1) \in \{0,1\}^{256 \times m_1 \hat{d}} \times \{0,1\}^{256 \times m_1 \hat{d}}.$$

Second round. Given the first challenge $(\mathfrak{R}_0, \mathfrak{R}_1)$, we compute the response $\vec{z}_3 := \vec{y}_3 + \mathfrak{R}\vec{s}_1 \in \mathbb{Z}^{256}$, where $\mathfrak{R} := \mathfrak{R}_0 - \mathfrak{R}_1$ and also $\vec{y}_3 := \mathsf{Coeffs}(\mathbf{y}_3), \vec{s}_1 = \mathsf{Coeffs}(\mathbf{s}_1)$. Then, we apply rejection sampling on $\vec{z}_3$. The second message and the corresponding challenge are:

$$a_2 := \vec{z}_3 \quad \text{and} \quad (\gamma_{i,j})_{i \in [\tau], j \in [256+\mathsf{n}+1]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2) \in \mathbb{Z}_{\hat{q}}^{\tau \times (256+\mathsf{n}+1)}.$$

Third round. The prover's goal now is to prove the following relations over $\mathbb{Z}_{\hat{q}}$:

$$\begin{cases} \vec{z}_3 = \vec{y}_3 + \mathfrak{R}\vec{s}_1 \\ P\vec{s}_1 = \vec{u} \\ \langle \vec{s}_1, \vec{s}_1 \rangle = \mathcal{B}^2 \end{cases}$$

where the terms in blue are secret and committed. As before, we deduce that proving the first equation is equivalent to proving that for $i \in [256]$, the constant coefficient of

$$\sigma(\mathbf{r}_i)^T \mathbf{s}_1 + \sigma(\mathbf{e}_i)^T \mathbf{y}_3 - z_{3,i} \text{ is equal to zero,}$$

where $\mathbf{r}_i$ is the polynomial vector such that its coefficient vector is the i-th row of $\mathfrak{R}$, and $\mathbf{e}_i$ is the polynomial vector so that $\mathsf{Coeffs}(\mathbf{e}_i)$ is a unit vector with its i-th coefficient being 1. Similarly, the second equation is equivalent to: for all $i \in [\mathsf{n}]$, the constant coefficient of:

$$\sigma(\mathbf{p}_i)^T \mathbf{s}_1 - \sigma(\mathbf{e}'_i)^T \mathbf{u} \text{ is equal to zero,}$$

where $\mathbf{p}_i$ is the polynomial vector with its coefficient vector being the i-th row of P, and $\mathbf{e}'_i$ is the polynomial vector so that $\mathsf{Coeffs}(\mathbf{e}'_i)$ is a unit vector with its i-th coefficient being 1. Obviously, the last equation is equivalent to $\sigma(\mathbf{s}_1)^T \mathbf{s}_1 - \mathcal{B}$ having the constant coefficient equal to zero. Hence, we compute

$$\begin{aligned} h_i := g_i &+ \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \mathbf{s}_1 + \sigma(\mathbf{e}_j)^T \mathbf{y}_3 - z_j \right) \\ &+ \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T \mathbf{s}_1 - \sigma(\mathbf{e}'_j)^T \mathbf{u} \right) \\ &+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\mathbf{s}_1)^T \mathbf{s}_1 - \mathcal{B} \right) \in \hat{\mathcal{R}}_{\hat{q}}. \end{aligned} \tag{26}$$

for $i = 1, 2, \ldots, \tau$. By definition of $\mathbf{g}$, the constant coefficients of $h_1, \ldots, h_\tau$ are all zeroes. The third message and the corresponding challenge are:

$$a_3 := (h_1, \ldots, h_\tau) \quad \text{and} \quad \boldsymbol{\mu} = (\mu_i)_{i \in [\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3) \in \hat{\mathcal{R}}_{\hat{q}}^{\tau}.$$

Fourth round. The goal of the prover is now to prove the τ quadratic equations (with automorphisms) over $\mathcal{R}_{\hat{q}}$ from (26). We now linear-combine them with the challenges μ_i:

$$\begin{aligned} 0 = \sum_{i=1}^{\tau} \mu_i \Bigg(&\sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \mathbf{s}_1 + \sigma(\mathbf{e}_j)^T \mathbf{y}_3 - z_j \right) + \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T \mathbf{s}_1 - \sigma(\mathbf{e}'_j)^T \mathbf{u} \right) \\ &+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\mathbf{s}_1)^T \mathbf{s}_1 - \mathcal{B} \right) + g_i - h_i \Bigg). \end{aligned} \tag{27}$$

Let us define

$$\mathbf{B} := \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix}, \quad \mathbf{t}_B := \begin{bmatrix} \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix}, \quad \mathbf{m} := \begin{bmatrix} \mathbf{y}_3 \\ \mathbf{g} \end{bmatrix} \in \hat{\mathcal{R}}_{\hat{q}}^{256/d+\tau} \quad \text{and} \quad \hat{\mathbf{s}} := \begin{bmatrix} \mathbf{s}_1 \\ \sigma(\mathbf{s}_1) \\ \mathbf{m} \\ \sigma(\mathbf{m}) \end{bmatrix}. \tag{28}$$

Then, the quadratic equation above can be written equivalently as

$$\hat{\mathbf{s}}^T \mathbf{D}_2 \hat{\mathbf{s}} + \mathbf{d}_1^T \hat{\mathbf{s}} + d_0 = 0$$

where

$$\begin{aligned} \mathbf{D}_2 &:= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \sum_{i=1}^{\tau} \mu_i \gamma_{i,256+\mathsf{n}+1} \cdot \mathbf{I}_{m_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \mathbf{d}_1 &:= \begin{bmatrix} \sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} \gamma_{i,j} \sigma(\mathbf{r}_j) + \sum_{j'=1}^{\mathsf{n}} \gamma_{i,256+j'} \sigma(\mathbf{p}_{j'}) \right) \\ \mathbf{0} \\ \sum_{i=1}^{\tau} \sum_{j=1}^{256} \mu_i \gamma_{i,j} \sigma(\mathbf{e}_j) \\ \boldsymbol{\mu} \\ \mathbf{0} \end{bmatrix} \\ d_0 &:= -\sum_{i=1}^{\tau} \mu_i \cdot \left(\sum_{j=1}^{256} z_j + \sum_{j'=1}^{\mathsf{n}} \sigma(\mathbf{e}'_{j'}) \mathbf{u} + \gamma_{i,256+\mathsf{n}+1} \mathcal{B} + h_i \right). \end{aligned} \tag{29}$$

Next, we run the sub-protocol for proving a single quadratic equation with automorphisms identically as before. First, we calculate the garbage term $f_1 = \hat{\mathbf{s}}^T \mathbf{R}_2 \mathbf{y} + \mathbf{y}^T \mathbf{R}_2 \hat{\mathbf{s}} + \mathbf{r}_1^T \mathbf{y}$, where $\mathbf{y}$ is defined as

$$\mathbf{y} := \begin{bmatrix} \mathbf{y}_1 \\ \sigma(\mathbf{y}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ -\sigma(\mathbf{B}\mathbf{y}_2) \end{bmatrix} \in \hat{\mathcal{R}}_{\hat{q}}^{2(m_1+256/d+\tau)}, \tag{30}$$

and the commitment $t = \mathbf{b}^T \mathbf{s}_2 + f_1$ to f_1. Then, we set $f_0 = \mathbf{y}^T \mathbf{R}_2 \mathbf{y} + \mathbf{b}^T \mathbf{y}_2$. Hence, the fourth message and the corresponding challenge are:

$$a_4 := (t, f_0) \quad \text{and} \quad c := \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4) \in \mathcal{C}.$$

Final round. Given a challenge c, we compute $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ for $i = 1, 2$ and apply rejection sampling on $\mathbf{z}_1, \mathbf{z}_2$. So, the prover's last message is $a_5 := (\mathbf{z}_1, \mathbf{z}_2)$ and thus the proof consists of

$$\pi := (a_1, a_2, a_3, a_4, a_5).$$

Verification algorithm. The verifier is given a proof π and recomputes the corresponding challenges as well as $\mathbf{D}_2, \mathbf{d}_1, d_0$. Denote

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \sigma(\mathbf{z}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \end{bmatrix}. \tag{31}$$

$\underline{\mathsf{LHC.Com}_i(\mathsf{crs}^{\mathsf{LHC}}, \mathbf{s}_i, \mathbf{y}_i)}$

$(\bar{\mathbf{e}}_{i,1}, \bar{\mathbf{e}}_{i,2}, \bar{\mathbf{e}}_{i,3}) \leftarrow S_{\eta_i}^{n_i} \times S_{\eta_i}^{m_i} \times S_{\eta_i}^{m_i}$
$\bar{\mathbf{t}}_{i,1} = \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{e}}_{i,1} + \bar{\mathbf{e}}_{i,2})$
$\bar{\mathbf{t}}_{i,2} = \bar{p} \cdot (\bar{\mathbf{B}}\bar{\mathbf{e}}_{i,1} + \bar{\mathbf{e}}_{i,3}) + \mathbf{s}_i$
$(\bar{\mathbf{f}}_{i,1}, \bar{\mathbf{f}}_{i,2}, \bar{\mathbf{f}}_{i,3}) \leftarrow D_{\bar{\mathfrak{s}}_i}^{n_i} \times D_{\bar{\mathfrak{s}}_i}^{m_i} \times D_{\bar{\mathfrak{s}}_i}^{m_i}$
$\bar{\mathbf{w}}_{i,1} = \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{f}}_{i,1} + \bar{\mathbf{f}}_{i,2})$
$\bar{\mathbf{w}}_{i,2} = \bar{p} \cdot (\bar{\mathbf{B}}\bar{\mathbf{f}}_{i,1} + \bar{\mathbf{f}}_{i,3}) + \mathbf{y}_i$
$\mathsf{com} = (\bar{\mathbf{t}}_{i,1}, \bar{\mathbf{t}}_{i,2}, \bar{\mathbf{w}}_{i,1}, \bar{\mathbf{w}}_{i,2})$
$\mathsf{st} = (\bar{\mathbf{e}}_{i,1}, \bar{\mathbf{e}}_{i,2}, \bar{\mathbf{e}}_{i,3}, \bar{\mathbf{f}}_{i,1}, \bar{\mathbf{f}}_{i,2}, \bar{\mathbf{f}}_{i,3})$
return $(\mathsf{com}, \mathsf{st})$

$\underline{\mathsf{LHC.Open}_i(\mathsf{crs}^{\mathsf{LHC}}, (\mathsf{com}, c), \mathsf{st})}$

for $j \in \{1,2,3\}$ **do**
  $\bar{\mathbf{z}}_{i,j} = \bar{\mathbf{f}}_{i,j} + c \cdot \bar{\mathbf{e}}_{i,j}$
$b \leftarrow \mathsf{Rej}([\bar{\mathbf{z}}_{i,1} \parallel \bar{\mathbf{z}}_{i,2} \parallel \bar{\mathbf{z}}_{i,3}],$
  $c \cdot [\bar{\mathbf{e}}_{i,1} \parallel \bar{\mathbf{e}}_{i,2} \parallel \bar{\mathbf{e}}_{i,3}], \bar{\mathfrak{s}}_i, \bar{M}_i)$
if $b = 0$ **then return** $\mathsf{op} = \bot$
else return $\mathsf{op} = [\bar{\mathbf{z}}_{i,1} \parallel \bar{\mathbf{z}}_{i,2} \parallel \bar{\mathbf{z}}_{i,3}]$

$\underline{\mathsf{LHC.Verify}_i(\mathsf{crs}^{\mathsf{LHC}}, (\mathsf{com}, c, (\mathbf{z}, \mathsf{op} \neq \bot))}$

$(\bar{\mathbf{t}}_{i,1}, \bar{\mathbf{t}}_{i,2}, \bar{\mathbf{w}}_{i,1}, \bar{\mathbf{w}}_{i,2}) \leftarrow \mathsf{com}$
$[\bar{\mathbf{z}}_{i,1} \parallel \bar{\mathbf{z}}_{i,2} \parallel \bar{\mathbf{z}}_{i,3}] \leftarrow \mathsf{op}$
if $\|\bar{\mathbf{z}}_{i,1}\| > \bar{\mathfrak{s}}_i \sqrt{2n_i d}$ **then return** 0
for $j \in \{2,3\}$ **do**
  if $\|\bar{\mathbf{z}}_{i,j}\| > \bar{\mathfrak{s}}_i \sqrt{2m_i \hat{d}}$ **then return** 0
$\bar{\mathbf{z}}_A = c\bar{\mathbf{t}}_{i,1} + \bar{\mathbf{w}}_{i,1} - \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{z}}_{i,1} + \bar{\mathbf{z}}_{i,2})$
$\bar{\mathbf{z}}_B = c\bar{\mathbf{t}}_{i,2} + \bar{\mathbf{w}}_{i,2} - \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{z}}_{i,1} + \bar{\mathbf{z}}_{i,3})$
if $\bar{\mathbf{z}}_A \neq \mathbf{0} \vee \bar{\mathbf{z}}_B \neq \mathbf{z}$ **then return** 0
else return 1

Fig. 11: Extractable linear homomorphic commitment from MLWE. Parts highlighted in gray correspond to the Katsumata transform.

Finally, the verifier outputs 1 if *all* the following relations hold:

$$\begin{cases} \|\mathbf{z}_1\| \overset{?}{\leq} \mathcal{B}_1, \|\mathbf{z}_2\| \overset{?}{\leq} \mathcal{B}_2, \|\vec{z}_3\| \overset{?}{\leq} \mathcal{B}_3, \\ \tilde{h}_i \overset{?}{=} 0 \text{ for } i \in [\tau], \\ \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 \overset{?}{=} \mathbf{w} + c\mathbf{t}_A, \\ \mathbf{z}^T\mathbf{D}_2\mathbf{z} + c\mathbf{d}_1^T\mathbf{z} + c^2 d_0 - (ct - \mathbf{b}^T\mathbf{z}_2) \overset{?}{=} f_0 \end{cases}$$

and 0 otherwise.

6.1 Katsumata Transform

In order to make the protocol multi-proof extractable, we apply the Katsumata transform [Kat21]. Recall that the protocol from [LNP22], as well as many prior linear-sized lattice-based proofs [ALS20; ENS20; BLS19; LNS21a; YAZ+19], follow the commit-and-prove approach. Namely, we first commit to $\mathbf{s}_1$ using randomness $\mathbf{s}_2$ and in the final round we send the masked openings $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ of $\mathbf{s}_i$ to prove knowledge of a commitment opening. Now, to make this protocol straight-extractable, we use an *extractable linear homomorphic commitment* (LHC), which can be seen as a linear homomorphic encryption scheme with pseudo-random public keys, and additionally encrypt both $\mathbf{s}_i$ and $\mathbf{y}_i$.

In this section, we recall the (simplified) LHC construction from Module-LWE [Kat21, Section 3.4][18]. Since we need to commit to both $(\mathbf{s}_1, \mathbf{y}_1)$ and $(\mathbf{s}_2, \mathbf{y}_2)$, we will need two instantiations. Fix $i \in \{1,2\}$. The common random string, i.e. the comitment key $\mathsf{crs}_i^{\mathsf{LHC}} = (\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i)$ contains two uniformly random matrices in $\mathcal{R}_{\hat{q}}^{m_i \times n_i} \times \mathcal{R}_{\hat{q}}^{m_i \times n_i}$. Next, we have three algorithms: $\mathsf{LHC.Com}_i, \mathsf{LHC.Open}_i$ and $\mathsf{LHC.Verify}_i$ which are described in Figure 11. In the following, let $\bar{p} < \hat{q}$ be an odd integer co-prime to $\hat{q}$.

We present the NIZK proof system $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}} := (\mathsf{Prove}_{\mathsf{Com}}^{\mathsf{H}}, \mathsf{Verify}_{\mathsf{Com}}^{\mathsf{H}})$ for relation R^{Com} in Figures 12 and 13. Intuitively, we take the protocol from above and additionally let the prover run $(\mathsf{com}_i, \mathsf{st}_i) \leftarrow$

[18] The reason why we do not apply the NTRU-type construction, as in [PK22], is that for efficiency we will pick relatively small ring dimension $\hat{d}$. In this case, the construction would rely on a *Module*-NTRU type assumption, which is still relatively uncommon and not well-studied.

$\mathsf{LHC.Com}_i(\mathsf{crs}_i^{\mathsf{LHC}}, \mathbf{s}_i, \mathbf{y}_i)$ for $i = 1, 2$. Hence, the first message now includes com_1 and com_2. Then, in the final round, the prover runs

$$\mathsf{op}_i \leftarrow \mathsf{LHC.Open}_i(\mathsf{crs}_i^{\mathsf{LHC}}, (\mathsf{com}_i, c), \mathsf{st}_i)$$

and appends op_1 and op_2 to the final message (unless one of them is $\perp$, then it restarts). Finally, the verifier additionally checks for $i = 1, 2$ that

$$\mathsf{LHC.Verify}(\mathsf{crs}_i^{\mathsf{LHC}}, (\mathsf{com}_i, c, \mathsf{op}_i \neq \perp)) \stackrel{?}{=} 1.$$

6.2 Security Analysis

In this subsection, we prove the key properties of $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$: correctness, zero-knowledge and multi-proof extractability.

6.2.1 Correctness

Lemma 6.1 (Correctness). *Let $\alpha_1, \alpha_2, \alpha_3, \bar{\alpha}_1, \bar{\alpha}_2 \in O(\sqrt{\lambda})$, $\mathsf{N}, d \in O(\lambda)$. Recall $\nu, \omega_{\max}$ from Section 5.1 and define the following standard deviations:*

$$\mathfrak{s}_1 := \alpha_1 \nu \mathcal{B}, \quad \mathfrak{s}_2 := \alpha_2 \nu \sqrt{m_2 \hat{d}}, \quad \mathfrak{s}_3 := \alpha_3 \mathcal{B} \omega_{\max}(\lambda)$$
$$\bar{\mathfrak{s}}_1 := \bar{\alpha}_1 \eta_1 \nu \sqrt{(n_1 + 2m_1)d}, \quad \bar{\mathfrak{s}}_2 := \bar{\alpha}_2 \eta_2 \nu \sqrt{(n_2 + 2m_2)d}$$

and the corresponding rejection rates for $i \in \{1, 2, 3\}$ and $j \in \{1, 2\}$:

$$M_i := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\alpha_i} + \frac{1}{2\alpha_i^2}\right), \quad \bar{M}_j := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\bar{\alpha}_j} + \frac{1}{2\bar{\alpha}_j^2}\right).$$

Then, $\Pi_{\mathsf{NIZK}}^{m} := (\mathsf{Prove}_{\mathit{ISIS}}^{\mathsf{H}}, \mathsf{Verify}_{\mathit{ISIS}}^{\mathsf{H}})$ for relation R^{Com} is correct.

Proof. The proof is almost identical to Lemma 5.5. Let us argue that the prover algorithm outputs $\perp$ with negligible probability. By Lemma 2.4 and the following inequalities (which hold with an overwhelming probability):

$$\|c\mathbf{s}_1\| \leq \mathcal{B}\nu, \quad \|c\mathbf{s}_2\| \leq \nu\sqrt{m_2\hat{d}}, \quad \|\mathfrak{R}\vec{s}_1\| \leq \mathcal{B} \cdot \omega_{\max},$$
$$\|c \cdot [\bar{\mathbf{e}}_{i,1} \parallel \bar{\mathbf{e}}_{i,2} \parallel \bar{\mathbf{e}}_{i,3}]\| \leq \eta_i \nu \sqrt{(n_i + 2m_i)d} \quad \text{for } i = 1, 2,$$

we deduce that in a single run the probability that $b_1 b_2 b_3 \bar{b}_1 \bar{b}_2 = 1$ is at least $(M_1 M_2 M_3 \bar{M}_1 \bar{M}_2)^{-1} - \mathsf{negl}(\lambda)$, and thus the probability that $\mathsf{Prove}_{\mathsf{Com}}^{\mathsf{H}}$ outputs $\perp$, i.e. makes exactly N unsuccessful attempts, is at most

$$\left(1 - \frac{1}{M_1 M_2 M_3 \bar{M}_1 \bar{M}_2} + \mathsf{negl}(\lambda)\right)^{\mathsf{N}} = \mathsf{negl}(\lambda)$$

since $M_1, M_2, M_3, \bar{M}_1, \bar{M}_2 \in O(1)$ and $\mathsf{N} = O(\lambda)$.

In terms of verification equations, by Lemma 2.2 for $t = \sqrt{2}$ and the union bound, the probability that one of the following conditions holds: (i) $\|\mathbf{z}_1\| > \mathfrak{s}_1\sqrt{2m_1\hat{d}}$, (ii) $\|\mathbf{z}_2\| > \mathfrak{s}_2\sqrt{2m_2\hat{d}}$, (iii) $\|\bar{\mathbf{z}}_{i,1}\| > \bar{\mathfrak{s}}_i\sqrt{2n_i\hat{d}}$ and (iv) $\|\bar{\mathbf{z}}_{i,j}\| > \bar{\mathfrak{s}}_i\sqrt{2m_i\hat{d}}$ for $i \in \{1, 2\}, j \in \{2, 3\}$, is negligible. We argue similarly for $\|\vec{z}_3\| > 1.7\mathfrak{s}_3\sqrt{256}$ where we apply Lemma 2.2 for $t = 1.7$ instead, as in Lemma 5.5. The rest of the verification checks hold by careful inspection of the protocol. □

$\underline{\mathsf{Prove}^{\mathsf{H}}_{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})}$

Input: $\mathsf{crs} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}, \bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2, \bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2)$, $\mathsf{x} = (P, \vec{u}, \mathcal{B})$, $\mathsf{w} = \vec{s}$

Output: proof π

$\mathsf{rst} = 0$ ▷ boolean which indicates if a restart is necessary

$\mathsf{idx} = 0$ ▷ keeps track how many restarts occured

$\mathbf{s}_1 = \mathbf{s}$ ▷ $\mathbf{s}$ is the polynomial vectors with coeff. $\vec{s}$

while $\mathsf{rst} = 0 \wedge \mathsf{idx} < \mathsf{N}$ **do**

  $\mathsf{idx} = \mathsf{idx} + 1$

  $\mathbf{s}_2 \leftarrow \chi^{m_2}$ ▷ generate commitment randomness

  $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$ ▷ Ajtai commitment to $\mathbf{s}_1$

  $\mathbf{y}_1 \leftarrow D^{m_1\hat{d}}_{\mathfrak{s}_1}, \mathbf{y}_2 \leftarrow D^{m_2\hat{d}}_{\mathfrak{s}_2}, \mathbf{y}_3 \leftarrow D^{256}_{\mathfrak{s}_3}$ ▷ sample masking vectors

  $\mathbf{g} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$ ▷ sample random polynomials with const. coeff. zero

  $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$ ▷ used to prove knowledge of a commitment opening

  $\mathbf{t}_y = \mathbf{B}_y\mathbf{s}_2 + \mathbf{y}_3$ ▷ commitment to $\mathbf{y}_3$

  $\mathbf{t}_g = \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}$ ▷ commitment to $\mathbf{g}$

  $(\mathsf{com}_1, \mathsf{st}_1) \leftarrow \mathsf{LHC.Com}_1((\bar{\mathbf{A}}_1, \bar{\mathbf{B}}_1), \mathbf{s}_1, \mathbf{y}_1)$ ▷ LHC to $\mathbf{s}_1$ and $\mathbf{y}_1$

  $(\mathsf{com}_2, \mathsf{st}_2) \leftarrow \mathsf{LHC.Com}_2((\bar{\mathbf{A}}_2, \bar{\mathbf{B}}_2), \mathbf{s}_2, \mathbf{y}_2)$ ▷ LHC to $\mathbf{s}_2$ and $\mathbf{y}_2$

  $a_1 = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathbf{v}, \mathsf{com}_1, \mathsf{com}_2)$ ▷ first message

  $(\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1) \in \{0,1\}^{256 \times m_1\hat{d}} \times \{0,1\}^{256 \times m_1\hat{d}}$ ▷ first challenge

  $\mathfrak{R} = \mathfrak{R}_0 - \mathfrak{R}_1$

  $\vec{s}_1 = \mathsf{Coeffs}(\mathbf{s}_1)$

  $\vec{y}_3 = \mathsf{Coeffs}(\mathbf{y}_3)$

  $\vec{z}_3 = \vec{y}_3 + \mathfrak{R}\vec{s}_1 \in \mathbb{Z}^{256}$ ▷ masked opening of $\mathfrak{R}\vec{s}_1$

  $b_3 \leftarrow \mathsf{Rej}\,(\vec{z}_3, \mathfrak{R}\vec{s}_1, \mathfrak{s}_3, M_3)$

  $a_2 = \vec{z}_3$ ▷ second message

  $(\gamma_{i,j})_{i\in[\tau], j\in[256+\mathsf{n}+1]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2) \in \mathbb{Z}_{\hat{q}}^{\tau \times (256+\mathsf{n}+1)}$ ▷ second challenge

  for $i \in [\tau]$ **do**

    compute h_i as in (26)

  $a_3 = \mathbf{h} = (h_1, \ldots, h_\tau)$ ▷ third message

  $(\mu_i)_{i\in[\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3) \in \hat{\mathcal{R}}_{\hat{q}}^{\tau}$ ▷ third challenge

  compute $\mathbf{B}, \mathbf{t}_B, \mathbf{m}, \hat{\mathbf{s}}, \mathbf{y}$ as in (28) and (30)

  compute $\mathbf{D}_2, \mathbf{d}_1, d_0$ as in (29)

  $f_1 = \mathbf{s}^T\mathbf{R}_2\mathbf{y} + \mathbf{y}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{y}$

  $f_0 = \mathbf{y}^T\mathbf{R}_2\mathbf{y} + \mathbf{b}^T\mathbf{y}_2$

  $t = \mathbf{b}^T\mathbf{s}_2 + f_1$ ▷ commitment to f_1

  $a_4 = (t, f_0)$ ▷ fourth message

  $c = \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4) \in \mathcal{C}$ ▷ fourth challenge

  for $i \in \{1, 2\}$ **do**

    $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ ▷ maked opening of $c\mathbf{s}_i$

    $b_i \leftarrow \mathsf{Rej}\,(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i, M_i)$ ▷ rejection sampling

    $\mathsf{op}_i \leftarrow \mathsf{LHC.Open}_i((\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i), (\mathsf{com}_i, c), \mathsf{st}_i)$ ▷ opening of the LHC

    if $\mathsf{op}_i = \perp$ **then** $\bar{b}_i = 0$

    else $\bar{b}_i = 1$

  $\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathsf{com}_1, \mathsf{com}_2, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2)$

  $\mathsf{rst} = b_1b_2b_3\bar{b}_1\bar{b}_2$ ▷ if all rejection steps passed then $\mathsf{rst} = 1$

if $\mathsf{rst} = 1$ **then return** π

else return $\perp$

Fig. 12: $\mathsf{Prove}^{\mathsf{H}}_{\mathsf{Com}}$ algorithm.

$\underline{\mathsf{Verify}^{\mathsf{H}}_{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \pi)}$

Input: $\mathsf{crs} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}, \bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2, \bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2)$, $\mathsf{x} = (P, \vec{u}, \mathcal{B})$, π

Output: bit b

1: parse $\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathsf{com}_1, \mathsf{com}_2, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2)$
2: $a_1 = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathbf{v}, \mathsf{com}_1, \mathsf{com}_2)$
3: $a_2 = \vec{z}_3$
4: $a_3 = \mathbf{h}$
5: $a_4 = (t, f_0)$
6: $(\mathfrak{R}_0, \mathfrak{R}_1) = \mathsf{H}(1, \mathsf{crs}, \mathsf{x}, a_1)$ ▷ recomputing the challenges
7: $(\gamma_{i,j})_{i\in[\tau], j\in[256+\mathsf{n}+1]} = \mathsf{H}(2, \mathsf{crs}, \mathsf{x}, a_1, a_2)$
8: $(\mu_i)_{i\in[\tau]} = \mathsf{H}(3, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3)$
9: $c = \mathsf{H}(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4)$
10: $\mathbf{B} = \begin{bmatrix}\mathbf{B}_y \\ \mathbf{B}_g\end{bmatrix}, \mathbf{t}_B = \begin{bmatrix}\mathbf{t}_y \\ \mathbf{t}_g\end{bmatrix}$
11: $\mathbf{z} = \begin{bmatrix}\mathbf{z}_1 \\ \sigma(\mathbf{z}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)\end{bmatrix}$ ▷ grouping the masked openings to $c\mathbf{s}_1$ and $c\mathbf{m}$
12: compute $\mathbf{D}_2, \mathbf{d}_1, d_0$ as in (20)
13: **if** one of the conditions below does not hold, then **return** 0:
  1. $\|\mathbf{z}_1\| \overset{?}{\leq} \mathcal{B}_1 := \mathfrak{s}_1\sqrt{2m_1\hat{d}}, \|\mathbf{z}_2\| \overset{?}{\leq} \mathcal{B}_2 := \mathfrak{s}_2\sqrt{2m_2\hat{d}}, \|\vec{z}_3\| \overset{?}{\leq} \mathcal{B}_3 := 1.7\sqrt{256}$
  2. $\tilde{h}_i \overset{?}{=} 0$ for $i \in [\tau]$
  3. $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 \overset{?}{=} \mathbf{w} + c\mathbf{t}_A$
  4. $\mathbf{z}^T\mathbf{D}_2\mathbf{z} + c\mathbf{d}_1^T\mathbf{z} + c^2d_0 - (ct - \mathbf{b}^T\mathbf{z}_2) \overset{?}{=} f_0$
  5. $\mathsf{LHC.Verify}_i((\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i), c, (\mathbf{z}_i, \mathsf{op}_i)) \overset{?}{=} 1$ for $i \in \{1, 2\}$
14: **else return** 1 ▷ all the verification checks passed

Fig. 13: $\mathsf{Verify}^{\mathsf{H}}_{\mathsf{Com}}$ algorithm.

6.2.2 Zero-Knowledge

Theorem 6.2 (Zero-Knowledge). *As before, let $\alpha_1, \alpha_2, \alpha_3, \bar{\alpha}_1, \bar{\alpha}_2 \in O(\sqrt{\lambda})$ and $\mathsf{N}, n, d \in O(\lambda)$. Define the following standard deviations:*

$$\mathfrak{s}_1 := \alpha_1\mathcal{B}\nu, \quad \mathfrak{s}_2 := \alpha_2\nu\sqrt{m_2\hat{d}}, \quad \mathfrak{s}_3 := \alpha_3\mathcal{B}\omega_{\max}(\lambda)$$
$$\bar{\mathfrak{s}}_1 := \bar{\alpha}_1\eta_1\nu\sqrt{(n_1+2m_1)d}, \quad \bar{\mathfrak{s}}_2 := \bar{\alpha}_2\eta_2\nu\sqrt{(n_2+2m_2)d}$$

and the corresponding rejection rates for $i \in \{1, 2, 3\}$ and $j \in \{1, 2\}$:

$$M_i := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\alpha_i} + \frac{1}{2\alpha_i^2}\right), \quad \bar{M}_j := \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\bar{\alpha}_j} + \frac{1}{2\bar{\alpha}_j^2}\right).$$

Then, $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ for relation R^{Com} is zero-knowledge under the $\mathsf{MLWE}_{2m_1,2m_1+n_1,\eta_1}$, $\mathsf{MLWE}_{2m_2,2m_2+n_2,\eta_2,\hat{q}}$ and $\mathsf{MLWE}_{n+256/\hat{d}+\tau+1,m_2,\chi,\hat{q}}$ assumptions.

Proof. The proof is almost identical to the one in Lemma 5.6 and we show it using a hybrid argument. Let $\mathcal{A}$ be a PPT algorithm, which makes at most Q_0 queries to H and Q_1 queries to $\mathsf{Prove}_{\mathsf{ISIS}}$, outputs 1 with probability ϵ. In each i-th game, we will introduce PPT algorithms $\mathcal{S}_0, \mathcal{S}_1$, which have a shared state, and denote ε_i to be the probability that $1 \leftarrow \mathcal{A}^{\mathcal{S}_0,\mathcal{S}_1}(\mathsf{crs})$.

Game_1: Here, $\mathcal{S}_0$ simulates the random oracle queries using lazy sampling and $\mathcal{S}_1$ runs $\mathsf{Prove}_{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$

$\underline{\mathsf{LHC.ZKSim}_i(\mathsf{crs}^{\mathsf{LHC}}, c, \mathbf{z}_i)}$

1: $(\bar{\mathbf{z}}_{i,1}, \bar{\mathbf{z}}_{i,2}, \bar{\mathbf{z}}_{i,3}) \leftarrow D_{\bar{\mathfrak{s}}_i}^{n_i} \times D_{\bar{\mathfrak{s}}_i}^{m_i} \times D_{\bar{\mathfrak{s}}_i}^{m_i}$
2: $(\bar{\mathbf{t}}_{i,1}, \bar{\mathbf{t}}_{i,2}) \leftarrow \mathcal{R}_{\hat{q}}^{m_i} \times \mathcal{R}_{\hat{q}}^{m_i}$
3: $\mathbf{w}_1 = -c\mathbf{t}_{i,1} + \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{z}}_1 + \bar{\mathbf{z}}_2)$
4: $\mathbf{w}_2 = -c\mathbf{t}_{i,2} + \bar{p} \cdot (\bar{\mathbf{A}}\bar{\mathbf{z}}_1 + \bar{\mathbf{z}}_3) + \mathbf{z}_i$
5: $\mathsf{com} = (\bar{\mathbf{t}}_{i,1}, \bar{\mathbf{t}}_{i,2}, \bar{\mathbf{w}}_{i,1}, \bar{\mathbf{w}}_{i,2})$
6: $u \leftarrow [0, 1)$
7: **if** $u > 1/\bar{M}_i$ **then** $\mathsf{op} = \bot$
8: **else** $\mathsf{op} = [\bar{\mathbf{z}}_{i,1} \parallel \bar{\mathbf{z}}_{i,2} \parallel \bar{\mathbf{z}}_{i,3}]$
9: **return** $(\mathsf{com}, \mathsf{op})$

Fig. 14: Simulator for LHC.

truthfully. By construction, $\epsilon_1 = \epsilon$.

Game_2: When $\mathcal{S}_1$ is queried, before executing Line 6 it first samples $c \leftarrow \mathcal{C}$. Then in Line 34, instead of calling the random oracle $\mathcal{S}_0$, it uses the challenge c generated at the very beginning. If $(4, \mathsf{crs}, \mathsf{x}, a_1, a_2, a_3, a_4)$ was already queried earlier to $\mathcal{S}_0$, then $\mathcal{S}_1$ aborts. Otherwise, it continues with the $\mathsf{Prove}_{\mathsf{Com}}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$ algorithm. Similarly as in Lemma 5.6, we argue that

$$|\epsilon_2 - \epsilon_1| \leq \mathsf{N}Q_1(Q_0 + \mathsf{N}Q_1)2^{-n\hat{d}} + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda).$$

Game_3: In this game, immediately after sampling $c \leftarrow \mathcal{C}$ and generating $\mathbf{y}_1, \mathbf{y}_2$, the algorithm directly computes $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ and runs

$$(\mathsf{com}_i, \mathsf{op}_i) \leftarrow \mathsf{LHC.ZKSim}_i((\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i), c, \mathbf{z}_i)$$

defined in Figure 14 for $i = 1, 2$ instead. Then, by the zero-knowledge property of the extractable linear homomorphic commitment [Kat21, Lemma 3.12], there exist PPT adversaries $\bar{\mathcal{A}}_1, \bar{\mathcal{A}}_2$ such that

$$|\epsilon_3 - \epsilon_2| \leq \mathsf{N} \cdot \left(\mathsf{Adv}^{\mathsf{MLWE}}_{2m_1, 2m_1+n_1, \eta_1}(\bar{\mathcal{A}}_1) + \mathsf{Adv}^{\mathsf{MLWE}}_{2m_2, 2m_2+n_2, \eta_2}(\bar{\mathcal{A}}_2) + \mathsf{negl}(\lambda)\right) = \mathsf{negl}(\lambda).$$

Game_4: Here, $\mathcal{S}_1$ directly computes $\mathbf{z}_i = \mathbf{y}_i + c\mathbf{s}_i$ for $i = 1, 2$ and $\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$. Further, $\mathcal{S}_1$ later calculates $f_0 := \mathbf{z}^T\mathbf{D}_2\mathbf{z} + c\mathbf{d}_1^T\mathbf{z} + c^2d_0 - \mathfrak{f}$, where $\mathfrak{f} := ct - \mathbf{b}^T\mathbf{z}_2$, instead. By the verification equations and simple rearrangement argument, we have $\epsilon_4 = \epsilon_3$.

Game_5: Now, $\mathcal{S}_1$ directly samples each $\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2\hat{d}}$ and sets $b_2 = 1$ with probability $1/M_2$. By Lemma 2.4, we have $|\epsilon_5 - \epsilon_4| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

Game_6: This time, $\mathcal{S}_1$ samples uniformly random $\boldsymbol{\nu} \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{n+256/\hat{d}+\tau+1}$ and computes the commitment the following way:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_y \\ \mathbf{t}_g \\ t \end{bmatrix} := \boldsymbol{\nu} + \begin{bmatrix} \mathbf{A}\mathbf{s}_1 \\ \mathbf{y}_3 \\ \mathbf{g} \\ f_1 \end{bmatrix}.$$

One can then construct a PPT adversary $\mathcal{A}^{\mathsf{MLWE}}$ which breaks Module-LWE with probability $|\epsilon_5 - \epsilon_4|/Q_1$. Hence,

$$|\epsilon_6 - \epsilon_5| \leq \mathsf{N} \cdot Q_1 \cdot \mathsf{Adv}^{\mathsf{MLWE}}_{n+256/\hat{d}+\tau+1, m_2, \chi}(\mathcal{A}^{\mathsf{MLWE}}) = \mathsf{negl}(\lambda).$$

Game_7: Here, $\mathcal{S}_1$ samples directly $(\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, t) \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{n+256/\hat{d}+\tau+1}$. Thus, $\epsilon_7 = \epsilon_6$.

Game_8: Now, $\mathcal{S}_1$ generates $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1\hat{d}}$ and sets $b_1 = 1$ with probability $1/M_1$. By Lemma 2.4, we have $|\epsilon_8 - \epsilon_7| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

Game_9: Here, $\mathcal{S}_1$ picks $\mathbf{h} \leftarrow \{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$ directly. Since earlier $\mathbf{g}$ was chosen uniformly at random from $\{x \in \hat{\mathcal{R}}_{\hat{q}} : \tilde{x} = 0\}^{\tau}$, we get $\epsilon_9 = \epsilon_8$.

Game_{10}: This time $\mathcal{S}_1$ samples $\vec{z}_3 \leftarrow D_{\mathfrak{s}_3}^{256}$ and sets $b_3 = 1$ with probability $1/M_3$. By Lemma 2.4, $|\epsilon_{10} - \epsilon_9| \leq \mathsf{N} \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$.

We observe that in Game_{10} $\mathcal{S}_1$ does not use the witness w (recall that $\mathcal{S}_0$ only simulates the random oracle H using lazy sampling). Hence, by the hybrid argument, $|\epsilon_{10} - \epsilon| = \mathsf{negl}(\lambda)$ which concludes the proof. □

6.2.3 Multi-Proof Extractability

We concentrate on proving multi-proof extractability. First, we provide basic intuition. Assume there is a PPT adversary which makes at most Q_{H} random oracle queries and outputs Q_s statement-proof pairs $(x_i, \pi_i)_{i\in[Q_s]}$ with non-negligible probability ε. Intuitively, we will use the property of *extractable* linear homomorphic commitments to extract a witness candidate w_i for each x_i from π_i. Then, for a fixed $i \in [Q_s]$, we will prove that the probability that w_i is *not* a valid witness for x_i must be negligible. Otherwise, we could run the knowledge extractor $\mathcal{E}$ from [LNP22, Appendix B] to obtain a witness w_i' for x_i. By nature of the [LNP22] protocol, both w_i and w_i' are parts of two relaxed opening (c.f. Definition 5.1) to the same ABDLOP commitment. Hence, by Lemma 5.2 we conclude that $w_i \neq w_i'$ and the knowledge extractor finds a valid Module-SIS solution.

Theorem 6.3 (Multi-Proof Extractability). *Let* $B := 8\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$ *and*

$$\hat{q} > \max\left(\mathcal{B}, 16m_1\hat{d}\mathcal{B}_3, \frac{2}{\omega_{\min}(\lambda)^2}\mathcal{B}_3^2\right).$$

Then, the NIZK *proof system* $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$ *for relation* R^{Com} *is multi-proof extractable under the* $\mathsf{MLWE}_{m_1,m_1+n_1,\eta_1}$, $\mathsf{MLWE}_{m_2,m_2+n_2,\eta_2}$ *and the* $\mathsf{MSIS}_{n,m_1+m_2,B}$ *assumptions.*

Proof. We start with crs indistinguishability. Namely, define the simulator $\mathcal{S}_{\mathsf{crs}}$ which first samples

$$\tilde{\mathsf{crs}} := (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}) \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_1} \times \hat{\mathcal{R}}_{\hat{q}}^{n\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{256/\hat{d}\times m_2} \times \hat{\mathcal{R}}_{\hat{q}}^{\tau\times m_2},$$
$$(\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2) \leftarrow \hat{\mathcal{R}}_{\hat{q}}^{m_1\times n_1} \times \hat{\mathcal{R}}_{\hat{q}}^{m_2\times n_2}.$$

It also generates $\bar{\mathbf{D}}_{i,1} \leftarrow S_{\eta_i}^{m_i\times m_i}, \bar{\mathbf{D}}_{i,2} \leftarrow S_{\eta_i}^{m_i\times n_i}$ and sets $\bar{\mathbf{B}}_i := \bar{\mathbf{D}}_{i,1}\bar{\mathbf{A}}_i + \bar{\mathbf{D}}_{i,2}$. Finally, it outputs

$$\tilde{\mathsf{crs}} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}, \bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2, \bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2)$$

and the trapdoor $\mathsf{td} = (\bar{\mathbf{D}}_{1,1}, \bar{\mathbf{D}}_{1,2}, \bar{\mathbf{D}}_{2,1}, \bar{\mathbf{D}}_{2,2})$. Then, $\tilde{\mathsf{crs}}$ is indistinguishable from random crs based on the $\mathsf{MLWE}_{m_1,n_1,\eta_1}$ and $\mathsf{MLWE}_{m_2,n_2,\eta_2}$ assumptions.

We now turn to proving straight-line extractability. Suppose there is a PPT adversary which makes at most Q_{H} random oracle queries and outputs Q_s statement-proof pairs $(x_i, \pi_i)_{i\in[Q_s]}$ with non-negligible probability $\varepsilon(\lambda)$. We start by formally defining the $\mathsf{Multi\text{-}Extract}$ algorithm in Figure 15. To provide more intuition on what the algorithm does, consider the interactive proof implicitly defined in Figure 12 by $\mathsf{Prove}_{\mathsf{Com}}$. Then, a valid transcript is of the form

$$\mathsf{tr} = \big((\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathsf{com}_1, \mathsf{com}_2), (\mathfrak{R}_0, \mathfrak{R}_1), \vec{z}_3, (\gamma_{i,j}), \mathbf{h}, (\mu_i), (t, f_0), c, (\mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2)\big)$$

where the challenges are highlighted in gray. One of the main observations is that if there also exists a valid transcript

$$\mathsf{tr}^* = \big((\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathsf{com}_1, \mathsf{com}_2), (\mathfrak{R}_0, \mathfrak{R}_1), \vec{z}_3, (\gamma_{i,j}), \mathbf{h}, (\mu_i), (t, f_0), c, (\mathbf{z}_1^*, \mathbf{z}_2^*, \mathsf{op}_1^*, \mathsf{op}_2^*)\big)$$

$\underline{\mathsf{Multi\text{-}Extract}(Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}}, \tau, \mathsf{x}, \pi)}$

Input: $Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}} := (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_y, \mathbf{B}_g, \mathbf{b}, \bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2, \bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2)$,
$\mathsf{td} := (\bar{\mathbf{D}}_{1,1}, \bar{\mathbf{D}}_{1,2}, \bar{\mathbf{D}}_{2,1}, \bar{\mathbf{D}}_{2,2}), \mathsf{x} := (P, \vec{u}), \pi$

Output: witness w, or a triple $(\perp, (\mathsf{x}, \pi, \mathsf{aux}))$, or $(\perp, \perp)$ (unsuccessful)

parse $\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathsf{com}_1, \mathsf{com}_2, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2)$

$\mathbf{B} = \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix}, \mathbf{t}_B = \begin{bmatrix} \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix}$

compute $\mathbf{D}_2, \mathbf{d}_1, d_0$ as in (20)

if one of the conditions below does not hold, then **return** 0:

$\mathsf{com}_1 = (\bar{\mathbf{t}}_{1,1}, \bar{\mathbf{t}}_{1,2}, \bar{\mathbf{w}}_{1,1}, \bar{\mathbf{w}}_{1,2})$

$\mathsf{com}_2 = (\bar{\mathbf{t}}_{2,1}, \bar{\mathbf{t}}_{2,2}, \bar{\mathbf{w}}_{2,1}, \bar{\mathbf{w}}_{2,2})$

$\mathsf{BadChall} = \{c\}$

$t = 0$

while $t \leq T_{\max} := \frac{\lambda \cdot 2Q_{\mathsf{H}}}{\varepsilon}$ **do**

  $b = 1$

  $c' \leftarrow \mathcal{C} \backslash \mathsf{BadChall}$

  for $i \in \{1, 2\}$ **do** $\mathbf{z}'_i = (c'\bar{\mathbf{t}}_{i,2} + \bar{\mathbf{w}}_{i,2}) - \bar{\mathbf{D}}_{i,1}(c'\bar{\mathbf{t}}_{i,1} + \bar{\mathbf{w}}_{i,1}) \bmod \bar{p}$

  $\mathbf{z}' = \begin{bmatrix} \mathbf{z}'_1 \\ \sigma(\mathbf{z}'_1) \\ c'\mathbf{t}_B - \mathbf{B}\mathbf{z}'_2 \\ \sigma(c'\mathbf{t}_B - \mathbf{B}\mathbf{z}'_2) \end{bmatrix}$

  $\mathsf{f}' := c't - \mathbf{b}^T \mathbf{z}'_2$

  $f'_0 = \mathbf{z}'^T \mathbf{D}_2 \mathbf{z}' + c'\mathbf{d}_1^T \mathbf{z}' + c'^2 d_0 - \mathsf{f}'$

  if one of the following conditions does not hold, **then** $b = 0$:

  - $\|\mathbf{z}'_1\| > \mathcal{B}_1 := 2\mathfrak{s}_1 \sqrt{2m_1 \hat{d}}$
  - $\|\mathbf{z}'_2\| > \mathcal{B}_2 := 2\mathfrak{s}_2 \sqrt{2m_2 \hat{d}}$
  - $\mathbf{A}_1 \mathbf{z}'_1 + \mathbf{A}_2 \mathbf{z}'_2 = \mathbf{w} + c'\mathbf{t}_A$
  - $f'_0 \neq f_0$

  if $b = 1$ **then**

    $c^* = c' - c$

    for $i \in \{1, 2\}$ **do**

      $\mathbf{s}_i^* = \frac{\mathbf{z}'_i - \mathbf{z}_i}{c' - c}$

    $\mathsf{w} = \vec{s}_1^* \in \mathbb{Z}_{\hat{q}}^{\mathsf{m}\hat{d}}$

    if $(\mathsf{x}, \mathsf{w}) \in R^{\mathsf{Com}}$ **then return** w

    else

      $\mathsf{aux} = (\mathbf{s}_1^*, \mathbf{s}_2^*, c^*)$

      return $(\perp, (\mathsf{x}, \pi, \mathsf{aux}))$

  else $\mathsf{BadChall} \leftarrow \mathsf{BadChall} \cup \{c'\}$

$t = t + 1$

return $(\perp, \perp)$

Fig. 15: Multi-Extract algorithm.

which has the same partial transcript as tr up to the last response, then we must have $\mathbf{z}_1 = \mathbf{z}_1^*$ and $\mathbf{z}_2 = \mathbf{z}_2^*$ assuming that the underlying encryption scheme (LHC) has no decryption error. Now, for a fixed $\tilde{\mathsf{crs}}$, statement x and a valid transcript $\mathsf{tr} := (\mathsf{pref\_tr}, (\mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2))$, where $\mathsf{pref\_tr}$ is the partial transcript which excludes the last message, define the following set:

$$\Gamma(x, \mathsf{tr}) := \left\{ c^* \in \mathcal{C} : \begin{array}{c} \exists (\mathbf{z}_1^*, \mathbf{z}_2^*, \mathsf{op}_1^*, \mathsf{op}_2^*) \text{ s.t. } \left(\mathsf{pref\_tr}, (\mathbf{z}_1^*, \mathbf{z}_2^*, \mathsf{op}_1^*, \mathsf{op}_2^*)\right) \\ \text{is a valid transcript} \end{array} \right\}.$$

At a high level, $\mathsf{Multi\text{-}Extract}$ is first given such an accepting transcript tr (technically it can manually produce one given π). It also defines a set of bad challenges as $\mathsf{BadChall} := \{c\}$. Then, it samples $c' \leftarrow \mathcal{C} \backslash \mathsf{BadChall}$, decrypts $(\bar{\mathbf{w}}_{i,1} + c'\bar{\mathbf{t}}_{i,1}, \bar{\mathbf{w}}_{i,2} + c'\bar{\mathbf{t}}_{i,2})$ to obtain some $\mathbf{z}'_i$, for $i = 1, 2$, and checks whether

$$\|\mathbf{z}'_1\| \overset{?}{\leq} \mathcal{B}_1 := \mathfrak{s}_1 \sqrt{2m_1 \hat{d}}, \quad \|\mathbf{z}'_2\| \overset{?}{\leq} \mathcal{B}_2 := \mathfrak{s}_2 \sqrt{2m_2 \hat{d}},$$
$$\mathbf{A}_1 \mathbf{z}'_1 + \mathbf{A}_2 \mathbf{z}'_2 \overset{?}{=} \mathbf{w} + c'\mathbf{t}_A, \quad \mathbf{z}'^T \mathbf{D}_2 \mathbf{z}' + c' \mathbf{d}_1^T \mathbf{z}' + c'^2 d_0 - (c' t - \mathbf{b}^T \mathbf{z}'_2) \overset{?}{=} f_0,$$

where

$$\mathbf{z}' := \begin{bmatrix} \mathbf{z}'_1 \\ \sigma(\mathbf{z}'_1) \\ c'\mathbf{t}_B - \mathbf{B}\mathbf{z}'_2 \\ \sigma(c'\mathbf{t}_B - \mathbf{B}\mathbf{z}'_2) \end{bmatrix}.$$

If not then the algorithm adds c' to $\mathsf{BadChall}$ and restarts the procedure.

Let $\mathsf{GoodChall}(\mathsf{x}, \mathsf{tr})$ be the set of all challenges $c' \in \mathcal{C}$ (including c) for which the procedure above succeeds. The first step in analysing the success probability of $\mathsf{Multi\text{-}Extract}$ will be to lower-bound the size of $\mathsf{GoodChall}(\mathsf{x}, \mathsf{tr})$. To begin with, one observes that $\Gamma(x, \mathsf{tr}) \subseteq \mathsf{GoodChall}(\mathsf{x}, \mathsf{tr})$. Hence, we focus on showing a lower bound on $\Gamma(x, \mathsf{tr})$ instead. To this end, we present the following lemma. Since it is almost identical to [PK22, Lemma 4.7], we omit the proof.

Lemma 6.4 ([PK22]). *Let $\mathcal{A}$ be a PPT adversary that makes at most Q_{H} random oracle queries and*

$$\Pr\left[\begin{array}{c} (\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}, \\ \{(\mathsf{x}_k, \pi_k)\}_{k \in [Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\tilde{\mathsf{crs}}) \end{array} : \forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathit{Com}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1 \right] \geq \varepsilon(\lambda).$$

Then, we have

$$\Pr\left[\begin{array}{c} (\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k \in [Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \tilde{\mathsf{crs}}) \end{array} : \begin{array}{c} \forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathit{Com}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1 \\ \wedge |\Gamma(\mathsf{x}_k, \mathsf{tr}_k)| \geq \frac{\varepsilon}{2Q_{\mathsf{H}}} \cdot |\mathcal{C}| \end{array} \right] \geq \frac{\varepsilon(\lambda)}{2} - \mathsf{negl}(\lambda)$$

where the transcript tr_k is constructed naturally from π_k using $\mathsf{Verify}^{\mathsf{H}}_{\mathit{Com}}$.

We use the lemma above to give an upper-bound on the following probability:

$$\varepsilon_{\mathsf{fail}} := \Pr\left[\begin{array}{c} (\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k \in [Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \tilde{\mathsf{crs}}) \end{array} : \begin{array}{c} \forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1, \\ \exists i \in [Q_s], (\bot, \bot) \leftarrow \mathsf{Multi\text{-}Extract}(\mathsf{input}_i) \end{array} \right]$$

where $\mathsf{input}_i := (Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}}, \tau, \mathsf{x}_i, \pi_i)$. Consider a fixed $i \in [Q_s]$. By Lemma 6.4, with probability at least $\varepsilon/2 - \mathsf{negl}(\lambda)$, for each $k \in [Q_s]$ we have $|\Gamma(\mathsf{x}_k, \mathsf{tr}_k)| \geq \frac{\varepsilon}{2Q_{\mathsf{H}}} \cdot |\mathcal{C}|$. Assuming this event occurs, then in each trial of the loop in Line 9 of $\mathsf{Multi\text{-}Extract}$, the probability that we pick $c' \in \mathsf{GoodChall}(\mathsf{x}_i, \mathsf{tr}_i)$ is at least $\varepsilon/(2Q_{\mathsf{H}})$. Hence, by repeating the procedure $T_{\max} = \lambda \cdot (2Q_{\mathsf{H}}/\varepsilon)$ times, the algorithm fails on each trial with negligible probability. Hence, by the union bound over i and the fact that $Q_s = \mathsf{poly}(\lambda)$, we deduce that

$$\varepsilon_{\mathsf{fail}} \leq \frac{\varepsilon}{2} + \mathsf{negl}(\lambda).$$

Next, we focus on upper-bounding

$$\varepsilon_{\mathsf{bind}} := \Pr\left[\begin{array}{c} (\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k \in [Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \tilde{\mathsf{crs}}) \end{array} : \begin{array}{c} \forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1, \\ \exists i \in [Q_s], (\bot, (\mathsf{x}, \pi, \mathsf{aux})) \leftarrow \mathsf{Multi\text{-}Extract}(\mathsf{input}_i) \end{array} \right]$$

where $\mathsf{input}_i := (Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}}, \tau, \mathsf{x}_i, \pi_i)$. To this end, we show Lemma 6.5 and conclude that $\varepsilon_{\mathsf{bind}} = \mathsf{negl}(\lambda)$. Finally, by simple calculation:

$$\Pr\left[\begin{array}{c}(\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \tilde{\mathsf{crs}})\end{array} : \begin{array}{c}\forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathsf{ISIS}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1, \\ \mathsf{w}_k \leftarrow \mathsf{Multi\text{-}Extract}(\mathsf{input}_k) \\ (\mathsf{x}_k, \mathsf{w}_k) \in R^{\mathsf{Com}}\end{array}\right] = \varepsilon - \varepsilon_{\mathsf{fail}} - \varepsilon_{\mathsf{sp}}$$
$$\leq \frac{\varepsilon}{2} - \mathsf{negl}(\lambda)$$

which concludes the proof. □

Lemma 6.5. *Let $\mathcal{A}$ be a PPT adversary that makes at most $Q_{\mathsf{H}} = \mathsf{poly}(\lambda)$ random oracle queries and*

$$\Pr\left[\begin{array}{c}(\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(\tilde{\mathsf{crs}})\end{array} : \forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathit{ISIS}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1\right] \geq \varepsilon(\lambda),$$

where $\varepsilon(\lambda)$ is non-negligible. Then, we have

$$\Pr\left[\begin{array}{c}(\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{x}_k, \pi_k)\}_{k\in[Q_s]} \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \tilde{\mathsf{crs}})\end{array} : \begin{array}{c}\forall k \in [Q_s], \mathsf{Verify}^{\mathsf{H}}_{\mathit{ISIS}}(\tilde{\mathsf{crs}}, \mathsf{x}_k, \pi_k) = 1 \wedge \\ \exists i, (\bot, (\mathsf{x}, \pi, \mathsf{aux})) \leftarrow \mathit{Multi\text{-}Extract}(\mathsf{input}_i)\end{array}\right] = \mathsf{negl}(\lambda)$$

where $\mathsf{input}_i := (Q_{\mathsf{H}}, Q_s, 1/\varepsilon, \tilde{\mathsf{crs}}, \tau, \mathsf{x}_i, \pi_i)$.

Proof. Suppose that the probability above is non-negligible and denote it as $\delta(\lambda)$. We consider a PPT algorithm $\mathcal{P}^*$, which given $(\tilde{\mathsf{crs}}, \mathsf{td})$, it essentially executes $\mathcal{A}$ to obtain Q_s statement-proof pairs $(\mathsf{x}_k, \pi_k)_{k\in[Q_s]}$, checks whether the proofs are valid, and then runs each $\mathsf{Multi\text{-}Extract}(\mathsf{input}_k)$ to check whether it outputs $(\bot, (\mathsf{x}, \pi, \mathsf{aux}))$. If not, then $\mathcal{P}^*$ aborts. Otherwise, $\mathcal{P}^*$ outputs the triple $(\mathsf{x}, \pi, \mathsf{aux})$. Let us also recall the meaning of the output of $\mathcal{P}^*$. Here, $\mathsf{x} = (P, \vec{u})$ is a statement, π is the corresponding proof from Section 6 written as:

$$\pi = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathsf{com}_1, \mathsf{com}_2, \vec{z}_3, \mathbf{h}, t, f_0, \vec{z}_3, \mathbf{h}, \mathbf{z}_1, \mathbf{z}_2, \mathsf{op}_1, \mathsf{op}_2) \tag{32}$$

and $\mathsf{aux} = (\mathbf{s}_1^*, \mathbf{s}_2^*, c^*)$ is the relaxed opening of $\mathbf{t}_A$ (c.f. Definition 5.1) such that $(\mathsf{x}, \vec{s}_1^*) \notin R^{\mathsf{Com}}$. Clearly, the probability that $\mathcal{P}^*$ outputs something is δ. Further, $\mathcal{P}^*$ runs in time $Q_s \cdot Q_{\mathsf{H}} \cdot \mathsf{poly}(\lambda)/\varepsilon$ and makes at most Q_{H} random oracle queries.

Now, we can use the knowledge extractor $\mathcal{E}$ from [LNP22, Appendix B]. Roughly speaking, it runs $\mathcal{P}^*$ once to obtain a valid triple $(\mathsf{x}, \pi, \mathsf{aux})$ by simulating the random oracle queries. Denote π as in (32). Then, for fixed random coins, fixed common reference string $\tilde{\mathsf{crs}}$ and the fixed statement x (where the latter two are included in the hash input of the Fiat-Shamir transformation in π), it re-runs $\mathcal{P}^*$ with different random oracle outputs in order to obtain additional valid triples of the form $(\mathsf{x}, \pi', \mathsf{aux}')$ where

$$\pi' = (\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \mathsf{com}_1, \mathsf{com}_2, \vec{z}_3', \mathbf{h}', t', f_0', \vec{z}_3', \mathbf{h}', \mathbf{z}_1', \mathbf{z}_2', \mathsf{op}_1', \mathsf{op}_2'). \tag{33}$$

Clearly, for $\mathsf{aux} := (\mathbf{s}_1^*, \mathbf{s}_2^*, c^*)$ and $\mathsf{aux}' = (\mathbf{s}_1', \mathbf{s}_2', c')$ we might have $\mathbf{s}_1^* \neq \mathbf{s}_1'$. However, in this case $\mathcal{E}$ would find two distinct relaxed openings to $\mathbf{t}_A$ and by Lemma 5.2 it would compute a valid solution to the Module-SIS problem. Hence, from now on we assume that every valid triple $(\mathsf{x}, \pi_i, \mathsf{aux}_i)$ extracted from $\mathcal{P}^*$ by $\mathcal{E}$ satisfies $\mathsf{aux}_i := (\mathbf{s}_1^*, \mathbf{s}_2^*, c_i)$

From [LNP22, Theorem B.7], we deduce that given access to $\mathcal{P}^*$, $\mathcal{E}$ runs in expected $Q_s \cdot Q_{\mathsf{H}}^2 \cdot \mathsf{poly}(\lambda)/\varepsilon$ time and with probability at least

$$\delta' := \delta - (Q_{\mathsf{H}} + 1)\left(\frac{2}{|\mathcal{C}|} + \hat{p}_1^{-\hat{d}/2} + \hat{p}_1^{-\tau} + 2^{-\lambda}\right) \tag{34}$$

outputs one of the following:

1. a triple $(\mathsf{x}, \mathsf{open}, \mathsf{aux})$, where

$$\mathsf{x} = (P, \vec{u}, \mathcal{B}), \quad \mathsf{open} := (\mathbf{s}_1, \mathbf{s}_2, c) \quad \text{and} \quad \mathsf{aux} := (\mathbf{s}_1^*, \mathbf{s}_2^*, c^*)$$

   such that $(x, \vec{s}_1) \in R^{\mathsf{Com}}, (x, \vec{s}_1^*) \notin R^{\mathsf{Com}}$ and both open and aux are relaxed openings of the same ABDLOP commitment $\mathbf{t}_A$,
2. a non-zero vector $\bar{\mathbf{s}} \in \hat{\mathcal{R}}_{\hat{q}}^{m_1+m_2}$ such that $\|\bar{\mathbf{s}}\| \leq B$ and

$$\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix} \bar{\mathbf{s}} = \mathbf{0}.$$

Clearly, in the second case $\mathcal{E}$ directly gets a valid Module-SIS solution so we only focus on the former one. Note that $(\mathsf{x}, \vec{s}_1^*) \notin R^{\mathsf{Com}}$ thus we must have $\vec{s}_1^* \neq \vec{s}_1$. Consequently, $\mathcal{E}$ has found two distinct relaxed openings to the same ABDLOP commitment and by Lemma 5.2, it can compute a valid Module-SIS solution. Hence, in either case $\mathcal{E}$ solves $\mathsf{MSIS}_{n,m_1+m_2,B}$.

To finish the proof, we apply the expected-time to strict-time transformation (c.f. Lemma 2.14) on $\mathcal{E}$. This gives us an algorithm $\mathcal{B}$ which solves $\mathsf{MSIS}_{n,m_1+m_2,B}$ with probability at least $\delta'/2$ and runs in strict time

$$Q_s \cdot Q_{\mathsf{H}}^2 \cdot \frac{\mathsf{poly}(\lambda)}{\varepsilon \cdot \delta'}.$$

Therefore, if δ is non-negligible then $\mathcal{B}$ runs in strict polynomial time and solves $\mathsf{MSIS}_{n,m_1+m_2,B}$ with non-negligible probability. This leads to a contradiction. □

7 The NTRU-ISIS$_f$ Assumption

Note that due to the preimage sampling procedure, the ISIS_f assumption is not falsifiable. In this section, we introduce the falsifiable version of the problem, called $\mathsf{NTRU\text{-}ISIS}_f$, where the challenger additionally has the NTRU trapdoor to produce short preimages.

Definition 7.1 (The $\mathsf{NTRU\text{-}ISIS}_f$ Problem). *Let $\mathsf{pp} := (q, d, m, N, k, \mathfrak{s}, \mathcal{B})$ be a tuple of functions of the security parameter λ. Consider any efficiently computable function $f : [N] \to \mathcal{R}_q$. The $\mathsf{NTRU\text{-}ISIS}_f$ assumption is defined by the experiment in Figure 16. For an adversary $\mathcal{A}$, we define*

$$\mathsf{Adv}_{\mathsf{pp}}^{\mathsf{NTRU\text{-}ISIS}_f}(\mathcal{A}) = \Pr[\mathsf{Exp}_{\mathsf{pp}}^{\mathsf{NTRU\text{-}ISIS}_f}(\mathcal{A}) \to 1].$$

The $\mathsf{NTRU\text{-}ISIS}_f^{\mathsf{pp}}$ assumption states that for every PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\mathsf{pp}}^{\mathsf{NTRU\text{-}ISIS}_f}(\mathcal{A})$ is negligible.

Similarly as in Section 3, we consider the interactive version of the $\mathsf{NTRU\text{-}ISIS}_f$ problem.

Definition 7.2 (The Interactive $\mathsf{NTRU\text{-}ISIS}_f$ Problem). *Define public parameters $\mathsf{pp} := (q, d, m, \ell_m, \ell_r, N, \mathfrak{s}, \mathcal{B}_s, \mathcal{B}_m)$ as a tuple of functions of the security parameter λ. Consider any efficiently computable function $f : [N] \to \mathcal{R}_q$. The $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ assumption is defined by the experiment in Figure 17. For an adversary $\mathcal{A}$, we define*

$$\mathsf{Adv}_{\mathsf{pp}}^{\mathsf{Int\text{-}NTRU\text{-}ISIS}_f}(\mathcal{A}) = \Pr[\mathsf{Exp}_{\mathsf{pp}}^{\mathsf{Int\text{-}NTRU\text{-}ISIS}_f}(\mathcal{A}) \to 1].$$

The $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f^{\mathsf{pp}}$ assumption states that for every PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\mathsf{pp}}^{\mathsf{Int\text{-}NTRU\text{-}ISIS}_f}(\mathcal{A})$ is negligible.

Using the proof strategy from Theorem 3.3, one can efficiently reduce $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ to $\mathsf{NTRU\text{-}ISIS}_f$. In the following, we only provide a proof sketch and highlight the parts where it differs from the proof of Theorem 3.3.

$\underline{\mathsf{Exp}^{\mathsf{NTRU\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A})}$

$(a_1, \mathbf{B}) \leftarrow \mathtt{NTRU.TrapGen}(q, d)$
$\mathbf{a}_2 \leftarrow \mathcal{R}_q^m$
$\mathbf{A} = \begin{bmatrix} a_1 & \mathbf{a}_2^T & 1 \end{bmatrix}$
for $i \in [k]$:
  $x_i \leftarrow [N]$
  $\mathbf{s}_i \leftarrow \mathtt{GSampler}(a_1, \mathbf{a}_2, \mathbf{B}, \mathfrak{s}, f(x_i))$
$(x^*, \mathbf{s}^*) \leftarrow \mathcal{A}(\mathbf{A}, (x_i, \mathbf{s}_i)_{i \in [k]})$
if $\left(\mathbf{A}\mathbf{s}^* = f(x^*)\right) \wedge \left(\|\mathbf{s}^*\| \leq \mathcal{B}\right) \wedge \left((x^*, \mathbf{s}^*) \notin \{(x_i, \mathbf{s}_i) : i \in [k]\}\right)$
  then return 1
else return 0

Fig. 16: The $\mathsf{NTRU\text{-}ISIS}_f$ experiment.

$\underline{\mathsf{Exp}^{\mathsf{Int\text{-}NTRU\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A})}$

$(a_1, \mathbf{B}) \leftarrow \mathtt{NTRU.TrapGen}(q, d)$
$\mathbf{a}_2 \leftarrow \mathcal{R}_q^m$
$\mathbf{A} = \begin{bmatrix} a_1 & \mathbf{a}_2^T & 1 \end{bmatrix}$
$\mathbf{C} \leftarrow \mathcal{R}_q^{1 \times (\ell_m + \ell_r)}$
$\mathcal{M} = \emptyset$
$(x^*, \mathbf{s}^*, \mathbf{m}^*, \mathbf{r}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{pre}}}(\mathbf{A}, \mathbf{C})$
if $\left(\mathbf{m}^* \notin \mathcal{M}\right) \wedge \left(\mathbf{A}\mathbf{s}^* = f(x^*) + \mathbf{C}\begin{bmatrix}\mathbf{m}^* \\ \mathbf{r}^*\end{bmatrix}\right) \wedge \left(\|\mathbf{s}^*\| \leq \mathcal{B}_s\right) \wedge \left(\|(\mathbf{m}^*, \mathbf{r}^*)\| \leq \mathcal{B}_m\right)$
  then return 1
else return 0

$\underline{\mathcal{O}_{\mathsf{pre}}(\mathbf{m}, \mathbf{r})}$

if $\|(\mathbf{m}, \mathbf{r})\| < \mathcal{B}_m$
  then return $\perp$
$x \leftarrow [N]$
$\mathbf{s} \leftarrow \mathtt{GSampler}\left(a_1, \mathbf{a}_2, \mathbf{B}, \mathfrak{s}, f(x) + \mathbf{C}\begin{bmatrix}\mathbf{m} \\ \mathbf{r}\end{bmatrix}\right)$
$\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
return $(x, \mathbf{s})$

Fig. 17: Interactive version of the $\mathsf{NTRU\text{-}ISIS}_f$ problem.

Theorem 7.3 ($\mathsf{Int\text{-}NTRU\text{-}ISIS}_f \implies \mathsf{NTRU\text{-}ISIS}_f$)**.** *Define the public parameters* $\mathsf{pp} := (q, d, m, \ell_m, \ell_r, N, \mathfrak{s}, \mathcal{B}_s, \mathcal{B}_m)$ *such that* $\mathcal{B}_m \geq 1, (2\kappa+1)^{m+2} > q$*, and*

$$M := \exp\left(1 + \frac{1}{2\alpha^2}\right) \quad \textit{and} \quad \varepsilon := 2\left(\frac{1+\epsilon}{1-\epsilon}\right)\exp\left(-2\alpha^2 \cdot \frac{\pi - 1}{\pi}\right)$$

where $\epsilon = 2^{-\delta}/(4d), \alpha \geq 1$ *and* $\kappa \in \mathbb{Z}_+$*. Suppose*

$$\mathfrak{s} \geq \max\left(1.17\sqrt{q} \cdot \eta'_\epsilon(\mathbb{Z}), \alpha\mathcal{B}_m\kappa d\sqrt{(\ell_m + \ell_r)(m+2)}\right).$$

Then, for every adversary $\mathcal{A}$ *which makes at most* Q *queries to* $\mathcal{O}_{\mathsf{pre}}$*, there are adversaries* $\mathcal{A}'$ *and* $\mathcal{B}$*, which run in time essentially identical to* $\mathcal{A}$ *and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{NTRU\text{-}ISIS}_f}_{\mathsf{pp}'}(\mathcal{A}') \geq \frac{1}{6Q}\mathsf{Adv}^{\mathsf{Int\text{-}NTRU\text{-}ISIS}_f}_{\mathsf{pp}}(\mathcal{A}) - \frac{\ell_m + \ell_r}{6Q}\mathsf{Adv}^{\mathsf{MLWE}}_{1,m+1,\kappa}(\mathcal{B}) - \frac{2^{-\lambda}}{6} - \frac{T_{\max}^2 Q}{12N} \\ - \frac{\ell_m}{3} \cdot \left(\frac{q}{(2\kappa+1)^{m+2}}\right)^d - \left(Q - \frac{2}{3}\right)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M} + 2^{-\delta+1}\right)\end{aligned}$$

where $\mathsf{pp}' := (q, d, m, N, T_{\max}Q, \mathfrak{s}, \mathcal{B} = \mathcal{B}_s + \mathcal{B}_m\kappa d\sqrt{(\ell_m + \ell_r)(m+2)})$ *and* $T_{\max}$ *satisfies* $\left(1 - \frac{1}{M}\right)^{T_{\max}} \leq 2^{-\lambda}$.

Proof. We closely follow the security games from the proof of Theorem 3.3 and explain the main differences. Namely, Game_1 is the interactive $\mathsf{NTRU\text{-}ISIS}_f$. In Game_2, we substitute the uniformly random $\mathbf{C}$ with $\mathbf{C} = \mathbf{AD}$

where $\mathbf{D} \leftarrow S_\kappa^{(m+2)\times(\ell_m+\ell_r)}$. Observe that now we do not use the leftover hash lemma anymore. Indeed, to see that the latter is pseudorandom under the $\mathsf{MLWE}_{1,m+1,\kappa}$ assumption, denote

$$\mathbf{D} := \begin{bmatrix} d_{1,1} \cdots d_{\ell_m+\ell_r,1} \\ \mathbf{d}_{1,2} \cdots \mathbf{d}_{\ell_m+\ell_r,2} \\ d_{1,3} \cdots d_{\ell_m+\ell_r,3} \end{bmatrix}.$$

Then,

$$\mathbf{C} = \begin{bmatrix} a_1 \ \mathbf{a}_2^T \ 1 \end{bmatrix} \mathbf{D} = \begin{bmatrix} a_1 d_{1,1} \cdots a_1 d_{\ell_m+\ell_r,1} \end{bmatrix} + \mathbf{u}$$

where $\mathbf{u} := \mathbf{a}_2^T \begin{bmatrix} \mathbf{d}_{2,1} \cdots \mathbf{d}_{2,\ell_m+\ell_r} \end{bmatrix} + \begin{bmatrix} d_{1,3} \cdots d_{\ell_m+\ell_r,3} \end{bmatrix}$. Hence, under the Module-LWE assumption, $\mathbf{u}$ is pseudorandom, and so is $\mathbf{C}$.

One can then define security games from 3 to 7 as before. For Game_6, we note that now there exists a polynomial vector $\mathbf{u} \in \Lambda_q^\perp(\mathbf{A})$ with coefficients between -2κ and 2κ, since $q^d < (2\kappa+1)^{(m+2)d}$. Hence, $\|\mathbf{u}\| \leq 2\kappa\sqrt{(m+2)d}$ and

$$\mathfrak{s} \geq \alpha\mathcal{B}_m\kappa d\sqrt{(\ell_m+\ell_r)(m+2)} \geq \kappa d\sqrt{(m+2)} \geq 2\kappa\sqrt{(m+2)d} \geq \|\mathbf{u}\|.$$

Thus, we can argue similarly as before that $\Pr[\mathsf{E}_6] \geq \frac{1}{2}\Pr[\mathsf{E}_5]$.

Clearly, Game_7 stays unchanged. As for Game_8, thanks to the structure of NTRU lattices, we do not need to exclude the case when $\mathfrak{s}$ is smaller than the smoothing parameter of $\Lambda_q^\perp(\mathbf{A})$. Now, in order to use Lemma 2.5, we need the fact that applying `GSampler` is statistically close to sampling preimages from the discrete Gaussian distribution, i.e. Lemma 2.7. Hence,

$$\Pr[\mathsf{E}_8] \geq \Pr[\mathsf{E}_7] - T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M} + 2^{-\delta+1}\right).$$

Next, to analyse Game_9, we need an upper-bound on the probability that $|S(\mathbf{D}, i)| = 1$ for fixed $i \in [\ell_m]$. Using the same techniques as in [Lyu12, Lemma 5.2], we can show that this happens with probability at most $\left(\frac{q}{(2\kappa+1)^{m+2}}\right)^d$. Hence,

$$\Pr[\mathsf{E}_9] \geq \frac{1}{3}\Pr[\mathsf{E}_8] - \frac{\ell_m}{3} \cdot \left(\frac{q}{(2\kappa+1)^{m+2}}\right)^d.$$

We can define Game_{10} as in the previous proof and deduce that

$$\Pr[\mathsf{E}_{10}] \geq \Pr[\mathsf{E}_9] - (Q-1)T_{\max}\left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M} + 2^{-\delta+1}\right).$$

Finally, the rest of the proof from Theorem 3.3 remains unchanged. □

8 Anonymous Credentials from NTRU-ISIS$_f$

This section focuses on building an efficient anonymous credentials scheme based on the newly defined assumption $\mathsf{NTRU\text{-}ISIS}_f$ where f is a binary encoding function (see Section 3.1.2). Namely, let $N = 2^{\mathsf{t}}$. Then, the function $f : [N] \to \mathcal{R}_q$ is defined as

$$f(x) := \mathsf{Coeffs}^{-1}(B \cdot \mathsf{enc}(x)) \in \mathcal{R}_q$$

where $\mathsf{enc}(x)$ is a binary decomposition of an integer $0 \leq x - 1 < 2^{\mathsf{t}}$ and $B \in \mathbb{Z}_q^{d\times\mathsf{t}}$ is a fixed matrix.

For convenience, we introduce the following notation which will be heavily used throughout this section. For a matrix $C \in \mathbb{Z}_q^{n\times lm}$ denoted as

$$C := \begin{bmatrix} C_1 \ C_2 \cdots C_l \end{bmatrix} \quad \text{where each } C_i \in \mathbb{Z}_q^{n\times m},$$

Parameter	Explanation
d	degree of the cyclotomic ring $\mathcal{R}$
$\hat{d}$	degree of the cyclotomic ring $\hat{\mathcal{R}}$ used in $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$
q	anonymous credentials modulus
$\hat{q}_1$	modulus for the proof system $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ divisible by q
$\hat{q}_2$	modulus for the proof system $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ divisible by q
N	domain size of the function f
t	$\log N$, length of the binary vector $\vec{u}$
h	length of the string output by H_{M}
l	number of attributes, we assume $d \vert l\mathsf{h}$
m	length of $\mathbf{a}_2$
ℓ_r	length of the randomness $\mathbf{r}$ generated by the holder
ℓ_m	$l\mathsf{h}/d$
ψ	infinity norm of $\mathbf{r}$ and the output of H_{M}
$\mathfrak{s}$	preimages are sampled from $D^{(m+2)d}_{\mathfrak{s}}$

Table 2: Overview of parameters and notation.

and a (possibly empty) subset $S = (s_1, \ldots, s_k) \subseteq [l]$, where $s_1 < \ldots < s_k$, we define:

$$C_S := \left[C_{s_1}\ C_{s_2} \cdots C_{s_k}\right] \in \mathbb{Z}_q^{n \times km}.$$

We introduce an identical notation to vectors and matrices over $\mathbb{Z}_q$ and $\mathcal{R}_q$. Further, we define $\bar{S} := [l] \backslash S$ to be the complement of S with respect to $[l]$.

8.1 Construction

We start with the summary of notation and parameters in Table 2, and define the building blocks, which were developed throughout the paper and will be used in our main construction:

- Function $f : [N] \to \mathcal{R}_q$ described in more detail in Section 3.1.2,
- The NIZK proof system $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}} := (\mathsf{Prove}^{\mathsf{H}_{\mathsf{ISIS}}}_s, \mathsf{Verify}^{\mathsf{H}_{\mathsf{ISIS}}}_s)$ (with a common random string $\mathsf{crs}_{\mathsf{ISIS}}$) described in Section 5 for the relation (12), where the proof system modulus is $\hat{q}_1$,
- The NIZK proof system $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}} := (\mathsf{Prove}^{\mathsf{H}_{\mathsf{Com}}}_{\mathsf{Com}}, \mathsf{Verify}^{\mathsf{H}_{\mathsf{Com}}}_{\mathsf{Com}})$ (with a common random string $\mathsf{crs}_{\mathsf{Com}}$) described in Section 6 for the relation (23), where the proof system modulus is $\hat{q}_2$,
- Four hash functions $\mathsf{H}_{\mathsf{crs}}, \mathsf{H}_{\mathsf{Com}}, \mathsf{H}_{\mathsf{ISIS}}, \mathsf{H}_{\mathsf{M}}$ modelled as random oracles in the security proof. The first one is a special hash function which we will only query on input 0 to obtain the crs for the underlying primitives, i.e.

  $$\mathsf{H}_{\mathsf{crs}}(0) = (\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{crs}_{\mathsf{Com}})$$

  where $\mathsf{crs}_{\mathsf{ISIS}}$ and $\mathsf{crs}_{\mathsf{Com}}$ are used by $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ respectively. Next, $\mathsf{H}_{\mathsf{Com}}, \mathsf{H}_{\mathsf{ISIS}}$ are used by the NIZK proof systems $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$. Finally, we apply H_{M} to hash attributes into a vector of length h with coefficients between $-\psi$ and ψ. As usual, one can use a single hash function instead of four by applying the domain separation technique.
- NTRU trapdoor sampling functions ($\texttt{NTRU.TrapGen}, \texttt{GSampler}$) defined in Section 2.1.

We present the $\mathsf{AnonCreds} = (\mathsf{AnonCreds.Init}, \mathsf{AnonCreds.Issue}, \mathsf{AnonCreds.Verify})$ construction in Figure 18, Figure 19 and Figure 20. We assume that all the involved parties retrieve $(\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{crs}_{\mathsf{Com}})$ by querying $\mathsf{H}_{\mathsf{crs}}(0)$.

Key generation. The key generation algorithm runs $(a_1, \mathbf{B}) \leftarrow \texttt{NTRU.TrapGen}(q, d)$ and samples $\mathbf{a_2} \leftarrow \mathcal{R}_q^m$. It further generates vectors $\mathbf{c}_0 \leftarrow \mathcal{R}_q^{\ell_m}$ and $\mathbf{c}_1 \leftarrow \mathcal{R}_q^{\ell_r}$. Then, we define the public key $\mathsf{ipk} = (a_1, \mathbf{a_2}, \mathbf{c}_0, \mathbf{c}_1)$ and secret key $\mathsf{isk} = \mathbf{B}$.

AnonCreds.Init
Input: number of attributes l
Output: $(\mathsf{ipk}, \mathsf{isk})$
1: $(a_1, \mathbf{B}) \leftarrow \texttt{NTRU.TrapGen}(q, d)$
2: $\mathbf{a_2} \leftarrow \mathcal{R}_q^m$
3: $(\mathbf{c_0}, \mathbf{c_1}) \leftarrow \mathcal{R}_q^{\ell_m} \times \mathcal{R}_q^{\ell_r}$
4: $\mathsf{ipk} \leftarrow (a_1, \mathbf{a_2}, \mathbf{c_0}, \mathbf{c_1})$
5: $\mathsf{isk} \leftarrow \mathbf{B}$
6: **return** $(\mathsf{ipk}, \mathsf{isk})$

Fig. 18: AnonCreds.Init algorithm.

Credential issuing. Next, we give an intuition on the issuing protocol. The holder $\mathcal{H}$ is given $\mathsf{ipk}, \mathsf{attrs} = (\mathsf{a}_1, \ldots, \mathsf{a}_\mathsf{l})$, and the set of indices $\mathsf{idx} \subseteq [l]$ for which the attributes in attrs are public. Then, it computes the polynomial vector

$$\mathbf{m} = \mathsf{Coeffs}^{-1}(\mathsf{H}_\mathsf{M}(\mathsf{a}_1), \ldots, \mathsf{H}_\mathsf{M}(\mathsf{a}_\mathsf{l})) \in \mathcal{R}_q^{\ell_m}. \tag{35}$$

Further, to commit to $\mathbf{m}$, $\mathcal{H}$ samples randomness $\mathbf{r} \leftarrow S_\psi^{\ell_r}$ and computes the commitment to $\mathbf{m}$ as

$$u := \mathbf{c}_0^T \mathbf{m} + \mathbf{c}_1^T \mathbf{r} \in \mathcal{R}_q.$$

The next step is to prove knowledge of a commitment opening of u. Since we want to use the proof system $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$, we need to translate this relation from $\mathcal{R}_q$ to $\mathbb{Z}_q$. Since $\mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}}$ (resp. $\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}$) are public (resp. hidden), we transform into a relation of the form (23) by defining:

$$\begin{aligned} P &:= [\mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)], \\ \vec{s} &:= \left(\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r})\right), \\ \vec{u} &:= \mathsf{Coeffs}(u) - \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}}. \end{aligned}$$

Then, using the properties of multiplication matrices we deduce that:

$$\begin{aligned} \mathsf{Coeffs}(u) &= \mathsf{rot}(\mathbf{c}_0^T)\mathsf{Coeffs}(\mathbf{m}) + \mathsf{rot}(\mathbf{c}_1^T)\mathsf{Coeffs}(\mathbf{r}) \\ &= \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}} + \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}} + \mathsf{rot}(\mathbf{c}_1^T)\mathsf{Coeffs}(\mathbf{r}) \end{aligned}$$

and thus:

$$\begin{aligned} P\vec{s} &= \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}} + \mathsf{rot}(\mathbf{c}_1^T)\mathsf{Coeffs}(\mathbf{r}) \\ &= \mathsf{Coeffs}(u) - \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}} \\ &= \vec{u}. \end{aligned}$$

Finally, since in $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$ we need to prove relations modulo $\hat{q}_1$, we prove instead:

$$\left(\frac{\hat{q}_1}{q} \cdot P\right) \vec{s} = \frac{\hat{q}_1}{q} \cdot \vec{u} \pmod{\hat{q}_1}.$$

Hence, the holder sends the commitment $u \in \mathcal{R}_q$ and proof π.

The issuer $\mathcal{I}$, given secret key isk, attributes attrs', the set of indices idx and the message (u, π) from $\mathcal{H}$, first verifies the proof π, where both P and $\vec{u}$ can be manually computed by $\mathcal{I}$. Further, it generates a uniformly random $x \leftarrow [N]$ and samples $\mathbf{s} \leftarrow \texttt{GSampler}\,(a_1, \mathbf{a}_2, \mathbf{B}, \mathfrak{s}, f(x) + u)$, which by definition satisfies

$$[a_1 | \mathbf{a}_2^T | 1]\mathbf{s} = f(x) + u.$$

Then, $\mathcal{I}$ outputs $(x, \mathbf{s})$ to $\mathcal{H}$. The holder then checks whether $\mathbf{s}$ is short and satisfies the equation above. If so, then the credential is a triple $\mathsf{cred} = (\mathbf{s}, \mathbf{r}, x)$.

$\mathsf{cred}/\perp \leftarrow \underline{\mathsf{AnonCreds.Issue}_{\mathcal{I},\mathcal{H}}\langle(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}), (\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx})\rangle}$

This protocol between issuer $\mathcal{I}$ and holder $\mathcal{H} = (\mathcal{H}_1, \mathcal{H}_2)$ proceeds as follows:

$\underline{\mathcal{H}_1(\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx})}$

1: parse $\mathsf{attrs} = (\mathsf{a}_1, \ldots, \mathsf{a}_\mathsf{l})$
2: $\mathbf{m} = \mathsf{Coeffs}^{-1}(\mathsf{H}_\mathsf{M}(\mathsf{a}_1), \ldots, \mathsf{H}_\mathsf{M}(\mathsf{a}_\mathsf{l})) \in \mathcal{R}_q^{\ell_m}$
3: $\mathbf{r} \leftarrow S_\psi^{\ell_r}$
4: $u = \mathbf{c}_0^T\mathbf{m} + \mathbf{c}_1^T\mathbf{r}$
5: $P = [\mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)] \in \mathbb{Z}_q^{d\times(|\mathsf{idx}|\cdot\mathsf{h}+\ell_r d)}$
6: $\vec{s} = \big(\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r})\big) \in \mathbb{Z}_q^{|\overline{\mathsf{idx}}|\cdot\mathsf{h}+\ell_r d}$
7: $\vec{u} = \mathsf{Coeffs}(u) - \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}}\mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}} \in \mathbb{Z}_q^d$
8: $\pi \leftarrow \mathsf{Prove}_{\mathsf{Com}}^{\mathsf{H}_{\mathsf{Com}}}\left(\mathsf{crs}_{\mathsf{Com}}, (\hat{q}_1/q \cdot P, \hat{q}_1/q \cdot \vec{u}, \psi\sqrt{\mathsf{h}\cdot|\overline{\mathsf{idx}}| + \ell_r d}), \vec{s}\right)$
9: **return** $(\rho_1 = (u, \pi), \mathsf{st}_\mathcal{H} = (\mathbf{m}, \mathbf{r}))$

$\underline{\mathcal{I}(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}, \rho_1)}$

1: parse $\rho_1 = (u, \pi^m)$, $\mathsf{attrs}' = (\mathsf{a_1}', \ldots, \mathsf{a_l}')$
2: $\mathbf{m}' = \mathsf{Coeffs}^{-1}(\mathsf{H}_\mathsf{M}(\mathsf{a_1}'), \ldots, \mathsf{H}_\mathsf{M}(\mathsf{a_l}')) \in \mathcal{R}_q^{\ell_m}$
3: $P = [\mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)] \in \mathbb{Z}_q^{d\times(|\mathsf{idx}|\cdot\mathsf{h}+\ell_r d)}$
4: $\vec{u} = \mathsf{Coeffs}(u) - \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}}\mathsf{Coeffs}(\mathbf{m}')_{\mathsf{idx}} \in \mathbb{Z}_q^d$
5: **if** $\mathsf{Verify}_{\mathsf{Com}}^{\mathsf{H}_{\mathsf{Com}}}\left(\mathsf{crs}_{\mathsf{Com}}, (\hat{q}_1/q \cdot P, \hat{q}_1/q \cdot \vec{u}, \psi\sqrt{\mathsf{h}\cdot|\overline{\mathsf{idx}}| + \ell_r d}), \pi\right) = 0$
6: **then return** $\perp$
7: $x \leftarrow [N]$
8: $\mathbf{s} \leftarrow \mathtt{GSampler}\,(a_1, \mathbf{a}_2, \mathbf{B}, \mathtt{s}, f(x) + u)$
9: **return** $\rho_2 = (\mathbf{s}, x)$

$\underline{\mathcal{H}_2(\mathsf{ipk}, \rho_2, \mathsf{st}_\mathcal{H})}$

1: parse $\rho_2 = (\mathbf{s}, x)$, $\mathsf{st}_\mathcal{H} = (\mathbf{m}, \mathbf{r})$
2: **if** $\left(\|\mathbf{s}\| > \mathtt{s}\sqrt{(m+2)d}\right) \vee \left(\|\mathbf{r}\| > \psi\sqrt{\ell_r d}\right) \vee \left([a_1|\mathbf{a}_2^T|1]\mathbf{s} \neq f(x) + \mathbf{c}_0^T\mathbf{m} + \mathbf{c}_1^T\mathbf{r}\right)$
3: **then return** $\perp$
4: **else return** $\mathsf{cred} = (\mathbf{s}, \mathbf{r}, x)$

Fig. 19: AnonCreds.Issue protocol.

Verification. We describe the protocol between the holder $\mathcal{H}(\mathsf{ipk}, \mathsf{attrs}, \mathsf{cred}, \mathsf{idx})$ and the verifier $\mathcal{V}$. Informally, the holder wants to prove knowledge of a short vector $\mathbf{s} \in \mathcal{R}_q^{m+2}$, binary polynomial $f(x) \in \mathsf{B}$, a short vector $\mathbf{r} \in \mathcal{R}_q^{\ell_r}$, and private parts of $\mathbf{m}$ defined as in (35) (dictated by indices idx of public attributes) which satisfy:

$$[a_1|\mathbf{a}_2^T|1]\mathbf{s} = f(x) + \mathbf{c}_0^T\mathbf{m} + \mathbf{c}_1^T\mathbf{r}.$$

Now, we need to transform these relations to the ones of the form (12) in order to use $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$. From the properties of multiplication matrices we get

$$\begin{aligned}
&\mathsf{rot}([a_1|\mathbf{a}_2^T|1])\mathsf{Coeffs}(\mathbf{s}) \\
&\quad = \mathsf{Coeffs}([a_1|\mathbf{a}_2^T|1]\mathbf{s}) \\
&\quad = \mathsf{Coeffs}(f(x)) + \mathsf{rot}(\mathbf{c}_0^T)\mathsf{Coeffs}(\mathbf{m}) + \mathsf{rot}(\mathbf{c}_1^T)\mathsf{Coeffs}(\mathbf{r}) \\
&\quad = \mathsf{Coeffs}(f(x)) + \mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}}\mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}} + \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}}\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}} + \mathsf{rot}(\mathbf{c}_1^T)\mathsf{Coeffs}(\mathbf{r}).
\end{aligned}$$

Hence, let us define

$$\begin{aligned}
&P := \mathsf{rot}([a_1|\mathbf{a}_2^T|1]), \quad C := [\mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mid \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)], \quad \vec{m} := \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}}, \\
&\vec{s} := \mathsf{Coeffs}(\mathbf{s}), \quad \vec{r} := (\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r})), \quad \vec{u} := \mathsf{enc}(x).
\end{aligned}$$

One can check that

$$P\vec{s} = B\vec{u} + C\begin{bmatrix}\vec{m}\\ \vec{r}\end{bmatrix}$$

$\langle \perp, 1/0 \rangle \leftarrow \mathsf{AnonCreds.Verify}_{\mathcal{H},\mathcal{V}}\langle(\mathsf{cred}, \mathsf{attrs}, \mathsf{idx}), (\mathsf{ipk}, \mathsf{attrs}', \mathsf{idx})\rangle$

The protocol between holder $\mathcal{H}$ and verifier $\mathcal{V}$:

$\mathcal{H}(\mathsf{ipk}, \mathsf{attrs}, \mathsf{cred}, \mathsf{idx})$

1: parse $\mathsf{attrs} = (\mathsf{a}_1, \ldots, \mathsf{a}_\mathsf{l})$, $\mathsf{cred} = (\mathbf{s}, \mathbf{r}, x)$
2: $\mathbf{m} = (\mathsf{H}_\mathsf{M}(\mathsf{a}_1), \ldots, \mathsf{H}_\mathsf{M}(\mathsf{a}_\mathsf{l})) \in \mathcal{R}_q^{\ell_m}$
3: $P = \mathsf{rot}([a_1 | \mathbf{a}_2^T | 1]) \in \mathbb{Z}_q^{d \times (m+2)d}$
4: $C = [\mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mid \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)] \in \mathbb{Z}_q^{d \times (\ell_m + \ell_r)d}$
5: $\vec{m} = \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}} \in \mathbb{Z}_q^{|\mathsf{idx}| \cdot \mathsf{h}}$
6: $\vec{s} = \mathsf{Coeffs}(\mathbf{s}) \in \mathbb{Z}_q^{(m+2)d}$
7: $\vec{r} = (\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r})) \in \mathbb{Z}_q^{|\overline{\mathsf{idx}}| \cdot \mathsf{h} + \ell_r d}$
8: $\vec{u} = \mathsf{Coeffs}(f(x)) \in \mathbb{Z}_q^d$
9: $\mathsf{Bounds} = \left(\mathfrak{s}\sqrt{(m+2)d}, \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}\right)$
10: $\pi \leftarrow \mathsf{Prove}_{\mathsf{ISIS}}^{\mathsf{H}_{\mathsf{ISIS}}}(\mathsf{crs}_{\mathsf{ISIS}}, (\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}, \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{idx}), (\vec{s}, \vec{r}, \vec{u}))$
11: **return** π

$\mathcal{V}(\mathsf{ipk}, \mathsf{attrs}', \mathsf{idx}, \pi)$

1: parse $\mathsf{attrs}' = (\mathsf{a_1}', \ldots, \mathsf{a_l}')$
2: $\mathbf{m}' = \mathsf{Coeffs}^{-1}(\mathsf{H}_\mathsf{M}(\mathsf{a_1}'), \ldots, \mathsf{H}_\mathsf{M}(\mathsf{a_l}')) \in \mathcal{R}_q^{\ell_m}$
3: $P = \mathsf{rot}([a_1 | \mathbf{a}_2^T | 1]) \in \mathbb{Z}_q^{d \times (m+2)d}$
4: $C = [\mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}} \mid \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}}} \mid \mathsf{rot}(\mathbf{c}_1^T)] \in \mathbb{Z}_q^{d \times (\ell_m + \ell_r)d}$
5: $\vec{m}' = \mathsf{Coeffs}(\mathbf{m}')_{\mathsf{idx}} \in \mathbb{Z}_q^{|\mathsf{idx}| \cdot \mathsf{h}}$
6: $\mathsf{Bounds} = \left(\mathfrak{s}\sqrt{(m+2)d}, \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}\right)$
7: **return** $\mathsf{Verify}_{\mathsf{ISIS}}^{\mathsf{H}_{\mathsf{ISIS}}}(\mathsf{crs}_{\mathsf{ISIS}}, (\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}', \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{idx}), \pi)$

Fig. 20: AnonCreds.Verify protocol.

and also $\|\vec{s}\| \leq \mathfrak{s}\sqrt{(m+2)d}$ with high probability, $\|\vec{r}\| \leq \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}$ and $\vec{u} \in \{0,1\}^\mathsf{t}$. Then, to obtain a relation of the form (12), we lift the equation from $\mathbb{Z}_q$ to $\mathbb{Z}_{\hat{q}_2}$ as before, i.e. we prove instead:

$$\left(\frac{\hat{q}_2}{q} \cdot P\right)\vec{s} = \left(\frac{\hat{q}_2}{q} \cdot B\right)\vec{u} + \left(\frac{\hat{q}_2}{q} \cdot C\right)\begin{bmatrix}\vec{m} \\ \vec{r}\end{bmatrix} \pmod{\hat{q}_2}.$$

Then, we end up with exactly the relation (12). Thus, the holder outputs the proof π from $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$ to the verifier $\mathcal{V}$ w.r.t. the statement

$$(\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}, \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{idx}),$$

where $\mathsf{Bounds} = \left(\mathfrak{s}\sqrt{(m+2)d}, \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}\right)$.

Finally, $\mathcal{V}$ given the public key ipk, set of attributes attrs', set of indices idx and a proof π, manually re-computes the statement $(\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}, \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{idx})$ and checks whether the proof π is valid.

8.2 Security Analysis

Lemma 8.1 (Correctness). *Let $\delta, d = O(\lambda)$, $\epsilon = 2^{-\delta}/(4d)$ and $\mathfrak{s} \geq 1.17\sqrt{q} \cdot \eta'_\epsilon(\mathbb{Z})$. If both $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$ and $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$ are correct, then* AnonCreds *is correct.*

Proof. We start by remarking that any honest execution of $\mathsf{AnonCreds.Init}$ will succeed since it only entails random sampling and an invocation of `NTRU.TrapGen` which, by Lemma 2.6, always succeeds.

We next show that given a successful execution of $\mathsf{AnonCreds.Init}$, any honest execution of $\mathsf{AnonCreds.Issue}$ succeeds with overwhelming probability (Equation 10): the first round $\mathcal{H}_1(\mathsf{ipk}, \mathsf{attrs}, \mathsf{idx})$ performed by the holder always succeeds since it only involves random sampling, algebraic operations and generation of a proof. Not to mention the fact that for the vector $\vec{s}$ generated honestly by $\mathcal{H}_1$, we must have

$$\|\vec{s}\| = \| \left(\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r})\right) \| \leq \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}.$$

Further, the round $\mathcal{I}(\mathsf{isk}, \mathsf{attrs}', \mathsf{idx}, \rho_1)$ executed by the issuer might fail if

$$\mathsf{Verify}_{\mathsf{Com}}^{\mathsf{H}_{\mathsf{Com}}}\left(\mathsf{crs}_{\mathsf{Com}}, (\hat{q}_1/q \cdot P, \hat{q}_1/q \cdot \vec{u}, \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}), \pi\right) = 0,$$

but by the correctness of $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$ (c.f. Lemma 6.1) this happens with negligible probability. Here, we used the fact that if both $\mathcal{H}$ and $\mathcal{I}$ are given attributes attrs and attrs', which agree on indices in idx, then we must have that $\vec{u}$ computed by $\mathcal{H}$, and $\vec{u}$ computed by $\mathcal{I}$, are identical. Finally, the last round $\mathcal{H}_2(\mathsf{ipk}, \rho_2, \mathsf{st}_{\mathcal{H}})$ performed by the holder may fail if $\|\mathbf{s}\| > \mathfrak{s}\sqrt{(m+2)d}$. Recall that $\mathbf{s} \in \mathcal{R}_q^{m+2}$ is generated such that

$$[a_1|\mathbf{a}_2^T|1]\mathbf{s} = f(x) + u.$$

Note that $\mathbf{s_2}$ is randomly sampled from a distribution statistically close to a discrete Gaussian distribution by Lemma 2.7. Consequently, by Lemma 2.3, $\mathbf{s}$ satisfies the condition $\|\mathbf{s}\| \leq \mathfrak{s}\sqrt{(m+2)d}$, with an overwhelming probability. Also, since $\mathbf{r} \leftarrow S_\psi^{\ell_r}$, we get $\|\mathbf{r}\| \leq \psi\sqrt{\ell_r d}$.

Finally we show that given a successful execution of $\mathsf{AnonCreds.Init}$ and of $\mathsf{AnonCreds.Issue}$, any honest execution of $\mathsf{AnonCreds.Verify}$ succeeds with overwhelming probability (Equation 11): the first round $\mathcal{H}(\mathsf{ipk}, \mathsf{attrs}, \mathsf{cred}, \mathsf{idx})$ executed by the holder always succeeds since it only involves random sampling, algebraic operations and generation of a proof. Similarly as before, the vector $\vec{r}$ computed honestly by $\mathcal{H}$ satisfies

$$\|\vec{r}\| = \|(\mathsf{Coeffs}(\mathbf{m})_{\overline{\mathsf{idx}}}, \mathsf{Coeffs}(\mathbf{r}))\| \leq \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}.$$

Furthermore, the round $\mathcal{V}(\mathsf{ipk}, \mathsf{attrs}', \mathsf{idx}, \pi)$ executed by the verifier might fail if $\mathsf{Verify}_{\mathsf{ISIS}}^{\mathsf{H}_{\mathsf{ISIS}}}(\mathsf{crs}_{\mathsf{ISIS}}, (\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}', \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{idx}), \pi) = 0$ but by the correctness of $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$ (c.f. Lemma 5.5) this happens with negligible probability. Similarly as before, we used the observation that if both $\mathcal{H}$ and $\mathcal{V}$ are given attributes attrs and attrs', which agree on indices in idx, then we must have $\vec{m}$ (computed by $\mathcal{H}$) is equal to $\vec{m}'$ (computed by $\mathcal{V}$). □

Theorem 8.2 (Anonymity). *Suppose $\Pi_{\mathsf{NIZK}}^{Com}$ and $\Pi_{\mathsf{NIZK}}^{ISIS}$ are zero-knowledge and $\mathsf{MLWE}_{1,\ell_r,\psi}$ assumption is hard. Then,* $\mathsf{AnonCreds}$ *is anonymous.*

Proof. Let $\mathcal{A}$ be a PPT adversary against the anonymity game. We prove the statement by introducing a sequence of games where we denote ϵ_i the probability that $\mathcal{A}$ outputs wins the i-th game.

Game_1: This is the standard anonymity game. By definition, the probability that $\mathcal{A}$ outputs $\mathsf{b}' = \mathsf{b}$ is ϵ_1.

Game_2: In this game, instead of running $\mathsf{Prove}_{\mathsf{ISIS}}^{\mathsf{H}_{\mathsf{ISIS}}}$, the challenger simulates the holder's behaviour in $\mathsf{AnonCreds.Verify}_{\mathcal{H},\mathcal{V}}$ using the zero-knowledge simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ of $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$. Namely, when $\mathcal{A}$ makes a random oracle query to $\mathsf{H}_{\mathsf{ISIS}}$, the challenger calls Sim_0. Furthermore, in the $\mathsf{AnonCreds.Verify}_{\mathcal{H},\mathcal{V}}$ protocol, the challenger now outputs a simulated proof $\pi_{\mathsf{b}} \leftarrow \mathsf{Sim}_1(\mathsf{crs}_{\mathsf{ISIS}}, (\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}', \hat{q}_2/q))$ to the adversary. Note that Sim_1 is only called once during the game.

One can naturally construct a PPT adversary $\mathcal{A}'$ which wins the zero-knowledge game of $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ with probability $|\epsilon_2 - \epsilon_1|$. Concretely, the adversary first programs the output of $\mathsf{H}_{\mathsf{crs}}(0)$ to use the $\mathsf{crs}_{\mathsf{ISIS}}$ provided by the zero-knowledge game. Next, it internally runs $\mathcal{A}$ and simulates the challenger's behaviour given access to either the oracles $(\mathsf{H}_{\mathsf{ISIS}}, \mathsf{Prove})$ or $(\mathsf{Sim}_0, \mathsf{Sim}_1)$. Note that by the verification checks made by the challenger, $\mathcal{A}$ queries only valid statements. Thus, by assumption we obtain:

$$|\epsilon_2 - \epsilon_1| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}}(\mathcal{A}') = \mathsf{negl}(\lambda).$$

Game_3: In this game, instead of running $\mathsf{Prove}^{\mathsf{H}_{\mathsf{Com}}}_{\mathsf{Com}}$ in $\mathsf{AnonCreds.Issue}$, the challenger simulates the proof π_{b} of a validity of a commitment u_{b} using the zero-knowledge simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ of $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$. Namely, when $\mathcal{A}$ makes random oracle query to $\mathsf{H}_{\mathsf{Com}}$, the challenger calls Sim_0. Further, the challenger computes simulated proofs $\pi_{\mathsf{b}} := \mathsf{Sim}_1(\mathsf{crs}_{\mathsf{Com}}, (P, \vec{u}))$. Finally, the challenger outputs $(u_{\mathsf{b}}, \pi_{\mathsf{b}})$ as the first message pair to $\mathcal{A}$.

Similarly as before, one can construct a PPT adversary $\mathcal{A}'$ which wins the zero-knowledge game of $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ with probability $|\epsilon_3 - \epsilon_2|$. Hence, by assumption on $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ we get

$$|\epsilon_3 - \epsilon_2| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}}(\mathcal{A}') = \mathsf{negl}(\lambda).$$

Game_4: Here, the challenger simulates the commitment u. Namely, $\mathcal{C}$ picks the commitment u' uniformly at random from $\mathcal{R}_q$ and sets $u := u' + \mathbf{c}_0^T \mathbf{m}$. Then, there exists a natural PPT adversary $\mathcal{A}^{\mathsf{MLWE}}$ which can distinguish $\mathbf{c}_1^T \mathbf{r}$ from a uniformly random polynomial u with probability at least $|\epsilon_4 - \epsilon_3|$, i.e. solves Module-LWE. Hence,

$$|\epsilon_4 - \epsilon_3| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{1,\ell_r,\psi}(\mathcal{A}^{\mathsf{MLWE}}) = \mathsf{negl}(\lambda).$$

Game_5: Now, the challenger directly samples u uniformly at random from $\mathcal{R}_q$. Clearly, we get $\epsilon_5 = \epsilon_4$. The key observation here is that in Game_5, the behaviour of the challenger is independent of the bit b. Hence, we conclude that $\epsilon_5 = 1/2$ and the statement holds by the hybrid argument. □

Theorem 8.3 (One-More Unforgeability). *Let $\mathsf{h} = O(\lambda)$. Suppose $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ is multi-proof extractable and the proof system modulus $\hat{q}_2$ for $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ satisfies*

$$\hat{q}_2 > \max\left(\mathfrak{s}\sqrt{(m+2)d}, \psi\sqrt{(\ell_m+\ell_r)d}, 16 m_1 \hat{d}\mathcal{B}_3, \frac{2}{\omega_{\min}(\lambda)^2}\mathcal{B}_3^2\right)$$

where m_1, m_2 are the parameters defined in Section 5. Then, $\mathsf{AnonCreds}$ is one-more unforgeable under the $\mathsf{Int\text{-}NTRU\text{-}ISIS}^{\mathsf{pp}}_f$ and $\mathsf{MSIS}_{n,m_1+m_2,\bar{\mathcal{B}}}$ assumptions where $\mathsf{pp} := (q, d, m, \ell_m, \ell_r, N, \mathfrak{s}, \mathfrak{s}\sqrt{(m+2)d}, \psi\sqrt{(\ell_m+\ell_r)d})$ and $\bar{\mathcal{B}} = 4\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$.

Proof. Suppose there is a PPT adversary $\mathcal{A}$ which wins the one-more unforgeability game with non-neglibile probability ϵ. Suppose $\mathcal{A}$ makes at most Q_{Issue} type queries and $Q_{\mathsf{H}_{\mathsf{ISIS}}}$, $Q_{\mathsf{H}_{\mathsf{Com}}}$, $Q_{\mathsf{H}_{\mathsf{M}}}$ queries to the random oracles $\mathsf{H}_{\mathsf{ISIS}}$, $\mathsf{H}_{\mathsf{Com}}$, and H_{M} respectively. Without loss of generality, assume that $\mathcal{A}$ never repeats a random oracle query.

Hybrid games. We prove the statement by introducing a sequence of games. In the following, let E_i be the event that $\mathcal{A}$ wins in Game_i and denote $\mathcal{C}_i$ as the challenger in Game_i.

Game_0: This is the standard one-more unforgeability game. Hence, by definition we have $\Pr[\mathsf{E}_0] = \epsilon$.

Game_1: At the beginning of the game, $\mathcal{C}_1$ samples $\vec{h}_j \leftarrow \{-\psi, -\psi+1, \ldots, \psi\}^{\mathsf{h}}$ for all $j \in [Q_{\mathsf{H}_{\mathsf{M}}}]$. When the adversary queries $\mathsf{a_j}'$ as the j-th random oracle query to H_{M}, then the challenger just outputs h_j. However, if some collision occurs, i.e. $\vec{h}_i = \vec{h}_j$ for distinct indices i, j, then $\mathcal{C}_1$ aborts. By the union bound, we have

$$\Pr[\mathsf{E}_1] \geq \Pr[\mathsf{E}_0] - \frac{|Q_{\mathsf{H}_{\mathsf{M}}}|^2}{(2\psi+1)^{\mathsf{h}}} = \Pr[\mathsf{E}_0] - \mathsf{negl}(\lambda).$$

Game_2: In this game, the challenger programs the simulated $\tilde{\mathsf{crs}}$ used for multi-proof extractability. Namely, recall that in Game_1 we had $\mathsf{H}_{\mathsf{crs}}(0) = (\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{crs}_{\mathsf{Com}})$. Here, the challenger $\mathcal{C}_2$ honestly generates $\mathsf{crs}^s_{\mathsf{NIZK}}$, but it also runs the CRS simulator $\mathcal{S}_{\mathsf{crs}}$ and obtains $(\tilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathcal{S}_{\mathsf{crs}}(1^\lambda)$. Then, it programs $\mathsf{H}_{\mathsf{crs}}(0) := (\mathsf{crs}_{\mathsf{ISIS}}, \tilde{\mathsf{crs}})$ and keeps td.

It is easy to see that Game_3 and Game_2 are indinstinguishable by the CRS indistinguishability property of the multi-proof extractability, i.e. one can construct a PPT adversary $\mathcal{A}^{\mathsf{crs}}$ such that

$$\Pr[\mathsf{E}_2] \geq \Pr[\mathsf{E}_1] - \mathsf{Adv}^{\mathsf{crs}}_{\Pi^m_{\mathsf{NIZK}}}(\mathcal{A}^{\mathsf{crs}}) = \Pr[\mathsf{E}_1] - \mathsf{negl}(\lambda).$$

Game_3: In this game, the challenger $\mathcal{C}_3$ uses the multi-proof extractability of $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ to extract witnesses for all proofs π sent by $\mathcal{A}$ as a part of issue queries. Namely, when $\mathcal{A}$ submits the j-th issue query $(\mathsf{attrs}'_j, \mathsf{idx}_j, u_j, \pi_j)$, where $j \in [Q_{\mathsf{Issue}}]$, the challenger additionally runs

$$\mathsf{w}_j \leftarrow \mathsf{Multi\text{-}Extract}\left(1^\lambda, Q_{\mathsf{H}_{\mathsf{Com}}}, Q_{\mathsf{Issue}}, 1/\Pr[\mathsf{E}_2], \mathsf{td}, \left(\hat{q}_1/q \cdot P_j, \hat{q}_1/q \cdot \vec{u}_j\right), \pi_j\right).$$

Let $\mathsf{Abort}_{\mathsf{extract}}$ be the event that $\mathsf{w}_j \notin R^{\mathsf{Com}}\left(\hat{q}_1/q \cdot P_j, \hat{q}_1/q \cdot \vec{u}_j\right)$ for some $j \in [Q_{\mathsf{Issue}}]$, where $P_j, \vec{u}_j$ are computed as in the honest execution of the issuer. If $\mathsf{Abort}_{\mathsf{extract}}$ occurs then $\mathcal{C}_4$ aborts the game and overwrites the adversary's forgery to be $\perp$. Otherwise, it proceeds as $\mathcal{C}_3$. Observe that $\mathcal{C}_4$ does not make any use of the extracted witnesses w_j. In the following, let $\mathbf{m}_j \in \mathcal{R}_q^{\ell_m}$ and $\mathbf{r}_j \in \mathcal{R}_q^{\ell_r}$ be the vectors which satisfy:

$$\mathsf{w}_j = \left(\mathsf{Coeffs}(\mathbf{m}_j)_{\overline{\mathsf{idx}_j}}, \mathsf{Coeffs}(\mathbf{r}_j)\right) \quad \text{and} \quad \mathsf{Coeffs}(\mathbf{m}_j)_{\mathsf{idx}_j} = \mathsf{Coeffs}(\mathbf{m}'_j)_{\mathsf{idx}_j}$$

where $\mathbf{m}'_j = \mathsf{Coeffs}^{-1}(\mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{j},1}{}'), \ldots, \mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{j},\mathsf{l}}{}'))$. In particular, we have

$$u_j = \mathbf{c}_0^T \mathbf{m}_j + \mathbf{c}_1^T \mathbf{r}_j \quad \text{for } j = 1, 2, \ldots, Q_{\mathsf{Issue}}.$$

The challenger stores the pairs $(\mathbf{m}_j, \mathbf{r}_j)_{j \in [Q_{\mathsf{Issue}}]}$. Intuitively, these pairs are the extracted message/randomness committed and sent from the holder to the issuer. From now on, we will informally say that $(\mathbf{m}_j, \mathbf{r}_j)$ is the j-th issuing query to $\mathcal{C}_3$, rather than (u_j, π_j). By definition we have

$$\|(\mathbf{m}_j, \mathbf{r}_j)\| = \left\| \left(\mathsf{Coeffs}(\mathbf{m}_j)_{\overline{\mathsf{idx}_j}}, \mathsf{Coeffs}(\mathbf{r}_j), \mathsf{Coeffs}(\mathbf{m}'_j)_{\mathsf{idx}_j}\right) \right\| \leq \psi\sqrt{(\ell_m + \ell_r)d}.$$

Arguing identically as in [PK22, Lemma 3.6] and assuming that $\Pr[\mathsf{E}_2]$ is non-negligible, the runtime of $\mathcal{C}_3$ is still $\mathsf{poly}(\lambda)$ and also

$$\Pr[\mathsf{E}_3] \geq \frac{1}{2}\Pr[\mathsf{E}_2] - \mathsf{negl}(\lambda).$$

Introducing a wrapper. Now, we provide a wrapper algorithm $\mathcal{P}^*$ for $\mathcal{A}$. Namely, $\mathcal{P}^*$ is given the $\mathsf{crs}_{\mathsf{ISIS}}$ for the proof system $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$, and runs $\mathcal{A}$ as a challenger for Game_3 (where $\mathsf{crs}_{\mathsf{ISIS}}$ is programmed as a part of $\mathsf{H}_{\mathsf{crs}}(0) := (\mathsf{crs}_{\mathsf{ISIS}}, \tilde{\mathsf{crs}})$). It all simulates all random oracle queries, apart from the one to $\mathsf{H}_{\mathsf{ISIS}}$, using lazy sampling.

Suppose that at the end $\mathcal{A}$ outputs a set $\mathcal{L} := \{\langle \mathsf{attrs}^*_i, \mathsf{idx}^*_i \rangle\}$. For each $i \in [Q_{\mathsf{Issue}} + 1]$, write $\mathsf{attrs}^*_i = (\mathsf{a}_{\mathsf{i},1}{}^*, \ldots, \mathsf{a}_{\mathsf{i},\mathsf{l}}{}^*)$ and define

$$\mathbf{m}^*_i := \mathsf{Coeffs}^{-1}(\mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{i},1}{}^*), \ldots, \mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{i},\mathsf{l}}{}^*)) \in \mathcal{R}_q^{\ell_m}. \tag{36}$$

We let the algorithm $\mathcal{P}^*$ find an index t which satisfies

$$\mathsf{Coeffs}(\mathbf{m}^*_t)_{\mathsf{idx}^*_t} \notin \left\{\mathsf{Coeffs}(\mathbf{m}_1)_{\mathsf{idx}^*_t}, \ldots, \mathsf{Coeffs}(\mathbf{m}_{Q_{\mathsf{Issue}}})_{\mathsf{idx}^*_t}\right\}. \tag{37}$$

If it cannot find one, then it outputs $(\perp, \perp, \perp, 0)$. Otherwise, the algorithm outputs a pair $(\mathcal{I}, y, v, \mathcal{Q})$ defined as follows. First, let

$$\mathsf{x} := (\hat{q}_2/q \cdot P, \hat{q}_2/q \cdot C, \vec{m}, \hat{q}_2/q \cdot B, \mathsf{Bounds}, \mathsf{aux} := \mathsf{idx}^*_t) \tag{38}$$

be the statement for relation (12) where

$$P := \mathsf{rot}([a_1|\mathbf{a}_2^T|1]), \quad C := [\mathsf{rot}(\mathbf{c}_0^T)_{\mathsf{idx}_t^*} \mid \mathsf{rot}(\mathbf{c}_0^T)_{\overline{\mathsf{idx}_t^*}} \mid \mathsf{rot}(\mathbf{c}_1^T)], \quad \vec{m} := \mathsf{Coeffs}(\mathbf{m})_{\mathsf{idx}_t}.$$

Also, let $\pi_t := (a_1, a_2, a_3, a_4, a_5)$ be the proof sent by the adversary $\mathcal{A}$ in $\mathsf{AnonCreds.Verify}_{\mathcal{A},\mathcal{P}^*}\langle(\cdot,\cdot,\cdot),(\mathsf{ipk},\mathsf{attrs}_t^*,\mathsf{idx}_t^*)\rangle$. Define $\mathcal{I} := (I_1, I_2, I_3, I_4, I_5)$ as

$$I_1 := (1, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, a_1), I_2 := (2, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, a_1, a_2), \ldots, I_5 := (5, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, a_1, a_2, a_3, a_4, a_5).$$

Then, it obtains the challenges by querying

$$\begin{aligned}
&(\mathfrak{R}_0, \mathfrak{R}_1) := \mathsf{H}_{\mathsf{ISIS}}(I_1),\\
&(\gamma_{i,j})_{i\in[\tau],j\in[256+d+3]} := \mathsf{H}_{\mathsf{ISIS}}(I_2),\\
&(\mu_1, \ldots, \mu_\tau) := \mathsf{H}_{\mathsf{ISIS}}(I_3),\\
&c := \mathsf{H}_{\mathsf{ISIS}}(I_4)
\end{aligned}$$

and defines

$$\mathsf{tr} := \left(\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, a_1, (\mathfrak{R}_0, \mathfrak{R}_1), a_2, (\gamma_{i,j}), a_3, (\mu_i), a_4, c, a_5\right), \quad \text{and} \quad v := V(\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, \mathsf{tr})$$

where $V(\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, \mathsf{tr}) = 1$ if tr is the accepting transcript for statement x with public parameters $\mathsf{crs}_{\mathsf{ISIS}}$, and 0 otherwise. This implies that $\mathcal{P}^*$ makes at most $Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4$ queries to $\mathsf{H}_{\mathsf{ISIS}}$. Finally, $\mathcal{Q} := (\mathbf{m}_j, \mathbf{r}_j)_{j\in[Q_{\mathsf{Issue}}]}$ and $\mathcal{P}^*$ outputs

$$(\mathcal{I}, \mathsf{tr}, v, \mathcal{Q}).$$

Also, given such a tuple from $\mathcal{P}^*$, where $v = 1$, we will denote:

$$\mathcal{I}(\vec{m}) := \vec{m} \quad \text{and} \quad \mathcal{I}(\mathsf{aux}) := \mathsf{aux} = \mathsf{idx}_t^*.$$

Below, we provide a few basic properties of the wrapper $\mathcal{P}^*$ to keep in mind for the remainer of the proof.

Lemma 8.4. $\Pr[(\mathcal{I}, \mathsf{tr}, v, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}}) \wedge v = 1] \geq \Pr[\mathsf{E}_3]$.

Proof. Suppose $\mathcal{A}$ wins Game_3. Namely, the set $\mathcal{L}$ is of size $Q_{\mathsf{Issue}} + 1$ such that (by definition of the one-more unforgeability game):

- for all $i \in [Q_{\mathsf{Issue}} + 1]$, $\langle\perp, 1\rangle \leftarrow \mathsf{AnonCreds.Verify}_{\mathcal{A},\mathcal{P}^*}\langle(\cdot,\cdot,\cdot),(\mathsf{ipk},\mathsf{attrs}_i^*,\mathsf{idx}_i^*)\rangle$,
- for each distinct pairs $\langle\mathsf{attrs}_i^*, \mathsf{idx}_i^*\rangle \in \mathcal{L}$ and $\langle\mathsf{attrs}_j^*, \mathsf{idx}_j^*\rangle \in \mathcal{L}$, there exists at least one index i^* in both idx_i^* and idx_j^* such that $\mathsf{attrs}_{i,\mathsf{i}^*} \neq \mathsf{attrs}_{j,\mathsf{i}^*}$.

We claim there exists an index $t \in [Q_{\mathsf{Issue}} + 1]$ such that (37). Suppose it is not the case. Then, by the pigeonhole principle there are two distinct indices $s, t \in [Q_{\mathsf{Issue}} + 1]$ and $j \in [Q_{\mathsf{Issue}}]$ such that:

$$\mathsf{Coeffs}(\mathbf{m}_s^*)_{\mathsf{idx}_s^*} = \mathsf{Coeffs}(\mathbf{m}_j)_{\mathsf{idx}_s^*} \quad \text{and} \quad \mathsf{Coeffs}(\mathbf{m}_t^*)_{\mathsf{idx}_t^*} = \mathsf{Coeffs}(\mathbf{m}_j)_{\mathsf{idx}_t^*}.$$

It implies that for all $i \in \mathsf{idx}_s^* \cap \mathsf{idx}_t^*$, $\mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{s},\mathsf{i}}{}^*) = \mathsf{H}_{\mathsf{M}}(\mathsf{a}_{\mathsf{t},\mathsf{i}}{}^*)$. Since we excluded the case of getting a hash collision in Game_1, we have $\mathsf{a}_{\mathsf{s},\mathsf{i}}{}^* = \mathsf{a}_{\mathsf{t},\mathsf{i}}{}^*$ which leads to the contradiction with the second winning condition of one-more unforgeability. Thus, $\mathcal{P}^*$ outputs $v = 1$. □

Lemma 8.5. *The following holds:*

$$\Pr\left[(\mathcal{I}, \mathsf{tr}, 1, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}}) \wedge \mathcal{I}(\vec{m}) \notin \{\mathsf{Coeffs}(\mathbf{m})_{\mathcal{I}(\mathsf{aux})} : \exists \mathbf{r} \in \mathcal{R}_q^{\ell_r} \ s.t.\ (\mathbf{m}, \mathbf{r}) \in \mathcal{Q}\}\right] = 1.$$

Proof. The statement can be shown in a similar manner as for Lemma 8.4. □

Next, we state crucial forking property of $\mathcal{P}^*$.

Black-box access to: $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$

1. Run $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows: relay the $Q_{\mathsf{H}_{\mathsf{ISIS}}}+4$ random oracle queries to the random oracle and record all query-response pairs. Obtain $(\mathcal{I}, \mathsf{tr}, v, \mathcal{Q})$. Set $i = I_4$ and let c_i be the response to query i.
2. If $v = 0$, abort and return $v = 0$. Else, $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$.
3. Repeat:
   (a) sample $c_i' \in \mathcal{C} \backslash \{c_i\}$ without replacement,
   (b) run $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows to obtain $(\mathcal{I}', \mathsf{tr}', v', \mathcal{Q}')$: answer the query to i with c_i' while answering all the other queries consistently if the query was performed by $\mathcal{P}^*$ in the previous run, and with fresh random values otherwise,
   (c) If the input for some j-th issuing query to $\mathcal{P}^*$ is of the form $(\mathbf{m}_j', \mathbf{r}_j')$ where $\vec{m} = \mathsf{Coeffs}(\mathbf{m}_j')_{\mathsf{aux}}$, then stop running $\mathcal{P}^*$ and directly go back to Step 3(a),
   until an additional challenge c_i' with $v' = 1$ and $I_K' = I_K$ has been found, or until all challenges $c_i' \in \mathcal{C}$ have been tried.
4. If the latter case occurs, output $v = 0$.
5. Otherwise, return $\big((\mathcal{I}, \mathsf{tr}, v, \mathcal{Q}), (\mathcal{I}', \mathsf{tr}', v', \mathcal{Q}')\big)$.

Fig. 21: Basic extractors $\mathcal{E}(\mathsf{crs}_{\mathsf{ISIS}})$ (defined with black text) and $\mathcal{E}'(\mathsf{crs}_{\mathsf{ISIS}})$ (defined with black and blue text). Here, $K \in \{1, 2, 3, 4\}$ is a fixed constant.

Lemma 8.6. *Consider the algorithms $\mathcal{E}$ and $\mathcal{E}'$ in Figure 21. Then,* $\Pr[0 \leftarrow \mathcal{E}(\mathsf{crs}_{\mathsf{ISIS}})] = \Pr[0 \leftarrow \mathcal{E}'(\mathsf{crs}_{\mathsf{ISIS}})]$.

Proof. Clearly, if $\mathcal{E}$ outputs $v = 0$ then so does $\mathcal{E}'$. Hence, suppose that when running $\mathcal{E}'$, in particular Step 3, for some j-th issuing query to $\mathcal{P}^*$, the input $(\mathbf{m}_j', \mathbf{r}_j')$ satisfies $\vec{m} = \mathsf{Coeffs}(\mathbf{m}_j')_{\mathsf{aux}}$. In this case, $\mathcal{E}'$ stops the execution of $\mathcal{P}^*$ and goes back to Step 3(a), while $\mathcal{E}$ continues running $\mathcal{P}^*$. Assume $\mathcal{P}^*$ returns $(\mathcal{I}', \mathsf{tr}', v', \mathcal{Q}')$ and $I_4' = I_4$. By our assumption on the j-th query,

$$\vec{m} \in \{\mathsf{Coeffs}(\mathbf{m}')_{\mathsf{aux}} : \exists \mathbf{r}' \in \mathcal{R}_q^{\ell_r} \text{ such that } (\mathbf{m}', \mathbf{r}') \in \mathcal{Q}'\}. \tag{39}$$

Thus, by Lemma 8.5 this implies $v' = 0$. Therefore, $\mathcal{E}$ would still go back to Step 3(a), identically as $\mathcal{E}'$. This concludes the proof. □

Even though we will not use this lemma as a black-box, its core idea will be applied throughout the extraction procedure. In particular, the key observation is that, by construction, $\mathcal{E}'$ after Step 2 never answers an issuing query for any $(\mathbf{m}_j', \mathbf{r}_j')$ such that $\vec{m} = \mathsf{Coeffs}(\mathbf{m}_j')_{\mathsf{aux}}$. This will be crucial when reducing to $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$. Obviously, the lemma can be applied similarly if we reprogram the random oracle for I_1, I_2 or I_3.

Defining the extractor. Since we will now analyse the protocol from Figures 9 and 10, let us denote for simplicity:

$$\hat{q} := \hat{q}_2, \quad P := \frac{\hat{q}}{q} \cdot P, \quad C := \frac{\hat{q}}{q} \cdot C, \quad B := \frac{\hat{q}}{q} \cdot B$$
$$\mathcal{B}_s := \mathfrak{s}\sqrt{(m+2)d}, \quad \mathcal{B}_r := \psi\sqrt{\mathsf{h} \cdot |\overline{\mathsf{idx}}| + \ell_r d}.$$

If we want to apply the proof system $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$, we define the variables used in Section 5 as follows [19]:

$$\mathsf{n} := d, \quad \mathsf{m} := (m+2)d + \hat{d}, \quad \ell_m := |\mathsf{idx}| \cdot \mathsf{h}, \quad \ell_r := (l - |\mathsf{idx}|) \cdot 2\lambda + \ell_r d + \hat{d}.$$

[19] Note that we increase the values of m and ℓ_r by $\hat{d}$ since instead of proving, e.g. $\|\vec{s}\| \leq \mathcal{B}_s$, we prove $\|\vec{s}\| = \mathcal{B}_s$ using the transformation described in Section 5.2.

Further, $\hat{q}$ is product of two primes $\hat{p}_1, \hat{p}_2$ as described in Section 5.1.

Intuitively, we will use the extraction algorithm from [LNP22, Appendix B] to extract from the prover $\mathcal{P}^*$. Namely, suppose in the first run $(\mathcal{I}, \mathsf{tr}, 1, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ and let $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$. Then, the extractor wants to obtain short $\mathbf{m}^*, \mathbf{r}^*$, a short vector $\mathbf{s}^*$ and $f(x^*) \in \mathsf{B}$ such that

$$\begin{bmatrix} a_1 & \mathbf{a}_2^T & 1 \end{bmatrix} \mathbf{s}^* = f(x^*) + \mathbf{c}_0^T \mathbf{m}^* + \mathbf{c}_1^T \mathbf{r}^* \quad \text{and} \quad \vec{m} = \mathsf{Coeffs}(\mathbf{m}^*)_{\mathsf{aux}}.$$

The extractor will be constructed similarly as $\mathcal{E}'$ in Lemma 8.6, i.e. when the extractor has to answer any issuing query for input $(\mathbf{m}, \mathbf{r})$[20], it stops running the current execution of $\mathcal{P}^*$ if $\vec{m} = \mathsf{Coeffs}(\mathbf{m})_{\mathsf{aux}}$. This way, we make sure that when our ideal extractor outputs $(\mathbf{m}^*, \mathbf{r}^*, \mathbf{s}^*, f(x^*))$ satisfying the conditions above, it never answered any issuing query on input of the form $(\mathbf{m}^*, \cdot)$ since $\vec{m} = \mathsf{Coeffs}(\mathbf{m}^*)_{\mathsf{aux}}$. This gives us a natural reduction to $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$.

We start by defining the extractor $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$, which is a $(Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4)$-query random oracle algorithm in Fig. 22. Informally, $\mathcal{E}_4$ finds the witness for a single quadratic equation with automorphism described in (18) by constructing a $(1, 1, 1, 3)$-tree of transcripts using the extraction from [AFK22].

Black-box access to: $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$

1. Run $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows: relay the $Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4$ random oracle queries to the random oracles and record all query-response pairs. Obtain $(\mathcal{I}, \mathsf{tr}, v, \mathcal{Q})$. Set $i = I_4$ and let c_i be the response to query i.
2. If $v = 0$, abort and return $v = 0$. Else, $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$.
3. Repeat:
   (a) sample $c_i' \in \mathcal{C} \backslash \{c_i\}$ without replacement,
   (b) run $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows to obtain $(\mathcal{I}', \mathsf{tr}', v', \mathcal{Q}')$: answer the query to i with c_i' while answering all the other queries consistently if the query was performed by $\mathcal{P}^*$ in the previous run, and with fresh random value otherwise,
   (c) If the input for some j-th issuing query to $\mathcal{P}^*$ is of the form $(\mathbf{m}_j', \mathbf{r}_j')$ where $\vec{m} = \mathsf{Coeffs}(\mathbf{m}_j')_{\mathsf{aux}}$, then stop running $\mathcal{P}^*$ and directly go back to Step 3(a),

   until either 2 additional challenges c_i' with $v' = 1$ and $I_4' = I_4$ have been found or until all challenges $c_i' \in \mathcal{C}$ have been tried.
4. If the latter case occurs, output $v = 0$.
5. For $i = 0, 1, 2$, $\mathsf{tr}_i := \left(\mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, a_1, (\mathfrak{R}_0, \mathfrak{R}_1), a_2, (\gamma_{i,j}), a_3, (\mu_i), a_4, c^{(i)}, a_5^{(i)}\right)$ be the extracted transcripts.
6. Parse $a_5^{(i)} := (\mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)})$ and a_1, a_2, a_3, a_4 as in the protocol description in Figure 10,
7. Define $\bar{c} := c^{(1)} - c^{(0)}, \bar{\mathbf{s}}_i = \frac{\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(0)}}{c^{(1)} - c^{(0)}}$ for $i = 1, 2$ and $\bar{\mathbf{m}} = (\bar{\mathbf{y}}_3, \bar{\mathbf{g}}) := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2$.
8. If $\mathbf{z}_i^{(1)} - c^{(1)}\bar{\mathbf{s}}_i \neq \mathbf{z}_i^{(2)} - c^{(2)}\bar{\mathbf{s}}_i$ for some $i = 1, 2$:
   – return $\mathcal{I}$, $\begin{bmatrix} \bar{c} \cdot \left(\begin{pmatrix} \mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)} \end{pmatrix} - \begin{pmatrix} c^{(2)} - c^{(1)} \end{pmatrix} \bar{\mathbf{s}}_1 \right) \\ \bar{c} \cdot \left(\begin{pmatrix} \mathbf{z}_2^{(2)} - \mathbf{z}_2^{(1)} \end{pmatrix} - \begin{pmatrix} c^{(2)} - c^{(1)} \end{pmatrix} \bar{\mathbf{s}}_2 \right) \end{bmatrix}$ as the MSIS solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
9. Otherwise, return $\mathcal{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.

Fig. 22: Subextractor $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$ as a $(Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4)$-query random oracle algorithm.

[20] Recall that the extractor, as an issuer, is only given its commitment u and proof π, but we use the multi-proof extractability property to extract the secret $\mathbf{m}, \mathbf{r}$.

Lemma 8.7. *For any* $\mathsf{crs}_{\mathsf{ISIS}}$*, the extractor* $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$ *makes an expected number of at most* $3+2Q_{\mathsf{H}_{ISIS}}$ *queries to* $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$*. Furthermore, for uniformly random* $\mathsf{crs}_{\mathsf{ISIS}}$*,* $\mathcal{E}_4$ *outputs* $v=1$ *with probability at least*

$$\Pr[\mathsf{E}_3] - (Q_{\mathsf{H}_{ISIS}}+1)\cdot\frac{2}{|\mathcal{C}|}.$$

Next, let $\mathcal{I}=(I_1,I_2,I_3,I_4,I_5)$ *be the index vector obtained by* $\mathcal{E}_4$ *in the first step and denote*

$$(\mathfrak{R}_0,\mathfrak{R}_1) := \mathsf{H}_1(I_1), \quad (\gamma_{i,j}) := \mathsf{H}_2(I_2), \quad (\mu_i) := \mathsf{H}_3(I_3).$$

as the corresponding (recorded) random oracle responses. Further, parse

$$I_4 := \left(4, \mathsf{crs}_{\mathsf{ISIS}}, x, a_1, (\mathfrak{R}_0,\mathfrak{R}_1), a_2, (\gamma_{i,j}), a_3, (\mu_i), a_4\right).$$

Then, conditioned on $v=1$*, the extractor either returns* $(\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}},\bar{\mathbf{r}},\bar{\mathbf{u}}), \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1}\times\mathcal{R}_q^{m_2}\times\mathcal{R}_q^{256/\hat{d}+\tau}\times\bar{\mathcal{C}}$ *which satisfies the following relations:*

$$\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_y\\ \mathbf{t}_g\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}_y\\ \mathbf{B}_g\end{bmatrix}\bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{y}}_3\\ \bar{\mathbf{g}}\end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \le 2\mathcal{B}_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \le 2\mathcal{B}_2$$

and[21]

$$\begin{aligned} 0 = \sum_{i=1}^{\tau}\mu_i\Bigg(&\sum_{j=1}^{256}\gamma_{i,j}\cdot\left(\sigma(\mathbf{r}_j)^T\bar{\mathbf{s}}_1 + \sigma(\mathbf{e}_j)^T\bar{\mathbf{y}}_3 - z_j\right)\\ &+\sum_{j=1}^{\mathsf{n}}\gamma_{i,256+j}\cdot\left(\sigma(\mathbf{p}_j)^T\bar{\mathbf{s}} - \sigma(\boldsymbol{\beta}_j)^T\bar{\mathbf{u}} - m_{C,j} - \sigma(\mathbf{c}_{r,j})^T\bar{\mathbf{r}}\right)\\ &+\gamma_{i,256+\mathsf{n}+1}\cdot\left(\sigma(\bar{\mathbf{s}})^T\bar{\mathbf{s}} - \mathcal{B}_s\right) + \gamma_{i,256+\mathsf{n}+2}\cdot\left(\sigma(\bar{\mathbf{r}})^T\bar{\mathbf{r}} - \mathcal{B}_r\right)\\ &+\gamma_{i,256+\mathsf{n}+3}\cdot\sigma(\bar{\mathbf{u}}-\mathbf{x})^T\bar{\mathbf{u}} + g_i - h_i\Bigg), \end{aligned} \tag{40}$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{\mathcal{B}}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$ *where* $\bar{\mathcal{B}} := 4\nu\sqrt{\mathcal{B}_1^2+\mathcal{B}_2^2}$.

Proof. We first focus on the probability that $\mathcal{E}_4$ outputs $v=0$. The first observation is that by Lemma 8.6, we can simply remove Step 3(c). Then, the statement holds by proving almost identically as in [LNP22, Lemma B.8]. □

Next, we define the subextractor $\mathcal{E}_3$ which is informally responsible for proving knowledge of a solution of multiple quadratic equations (41). We present the subextractor in Fig. 23. Here, $\mathcal{E}_3$ uses the early abort feature of $\mathcal{E}_4$, as shown in [AFK22]. That is, $\mathcal{E}_4$ computes the index vector $\mathcal{I}$ by running $\mathcal{P}^*$ as the first step. This allows the executions in the repeat loop of $\mathcal{E}_3$ to abort right after a single run of $\mathcal{P}^*$ if $I_3' \neq I_3$. This allows to have the runtime of the extractor to be linear in the number of random oracle queries $Q_{\mathsf{H}_{\mathsf{ISIS}}}$.

The main intuition is that by running $\mathcal{E}_4$ once, $\mathcal{E}_3$ obtains a candidate witness w (or a Module-SIS solution, but we ignore the latter case). Since the ABDLOP commitment $(\mathbf{t}_A,\mathbf{t}_y,\mathbf{t}_g)$ to the witness was sent before getting (μ_i) and the commitment itself is binding, this means that the extracted witness must be *independent* of the challenges $\mu_1,\dots,\mu_\tau$. Therefore, if for some index $i\in[\tau]$, (41) does not hold, then only with probability at most $\hat{p}_1^{-\hat{d}/2}$ Equation 40, which is the relation that w satisfies by construction of $\mathcal{E}_4$, is true. Formally, the proof simply combines Lemma 8.6 with [LNP22, Lemma B.9] so we refer to [LNP22] for more details.

[21] See Equation 18 for more intuition on the following equation.

Black-box access to: $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$

1. Run $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness $\rho \leftarrow R$ as follows: relay the $Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4$ random oracle queries to the random oracles and record all query-response pairs. Obtain $(\mathcal{I}, y, v)$. Set $i = I_3$ and let $(\mu_1, \ldots, \mu_\tau)$ be the response to query i.
2. If $v = 0$, abort and return $v = 0$. Else, $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$.
3. Repeat:
   (a) sample $(\mu'_1, \ldots, \mu'_\tau) \leftarrow \hat{\mathcal{R}}^\tau_{\hat{q}}$,
   (b) run $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows to obtain $(\mathcal{I}', y', v')$, aborting right after (or during) the initial run of $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ if one of the following conditions holds:
   – $I'_3 \neq I_3$,
   – the input for some j-th issuing query to $\mathcal{P}^*$ is of the form $(\mathbf{m}'_j, \mathbf{r}'_j)$ where $\vec{m} = \mathsf{Coeffs}(\mathbf{m}'_j)_{\mathsf{aux}}$, answer the query to i with $(\mu'_1, \ldots, \mu'_\tau)$ while answering all the other queries consistently if the query was performed by $\mathcal{E}_4$ in the previous run, and with fresh random value otherwise,
   until a challenge $(\mu'_1, \ldots, \mu'_\tau)$ with $v' = 1$ and $I'_3 = I_3$ has been found.
4. If y (resp. y') is a MSIS solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathcal{I}$, y (resp. y') and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}_3, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}'_3, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathcal{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the MSIS solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. Parse $\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}}, \bar{\mathbf{r}}, \bar{\mathbf{u}})$, $(\mathfrak{R}_0, \mathfrak{R}_1) := \mathsf{H}_1(I_1), (\gamma_{i,j}) := \mathsf{H}_2(I_2)$ and $I_3 := (3, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, \mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3, h_1, \ldots, h_\tau)$.
8. If for $i = 1, 2, \ldots, \tau$ we have

$$
\begin{aligned}
h_i = g_i &+ \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + \sigma(\mathbf{e}_j)^T \bar{\mathbf{y}}_3 - z_{3,j} \right) \\
&+ \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T \bar{\mathbf{s}} - \sigma(\boldsymbol{\beta}_j)^T \bar{\mathbf{u}} - m_{C,j} - \sigma(\mathbf{c}_{r,j})^T \bar{\mathbf{r}} \right) \\
&+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\bar{\mathbf{s}})^T \bar{\mathbf{s}} - \mathcal{B}_s \right) + \gamma_{i,256+\mathsf{n}+2} \cdot \left(\sigma(\bar{\mathbf{r}})^T \bar{\mathbf{r}} - \mathcal{B}_r \right) \\
&+ \gamma_{i,256+\mathsf{n}+3} \cdot \sigma(\bar{\mathbf{u}} - \mathbf{x})^T \bar{\mathbf{u}} \in \hat{\mathcal{R}}_{\hat{q}}.
\end{aligned}
$$

   return $\mathcal{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
9. Otherwise, return $v = 0$.

Fig. 23: Subextractor $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$ as a $(Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4)$-query random oracle algorithm.

Lemma 8.8. *For any* $\mathsf{crs}_{\mathsf{ISIS}}$*, the extractor* $\mathcal{E}_3$ *makes an expected number of at most* $6 + 5Q_{\mathsf{H}_{ISIS}}$ *queries to* $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$*. Also, for uniformly random* $\mathsf{crs}_{\mathsf{ISIS}}$*,* $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$ *outputs* $v = 1$ *with probability at least*

$$\Pr[\mathsf{E}_3] - (Q_{\mathsf{H}_{ISIS}} + 1) \cdot \left(\frac{2}{|\mathcal{C}|} + \hat{p}_1^{-\hat{d}/2} \right).$$

Next, let $\mathcal{I} = (I_1, I_2, I_3, I_4, I_5)$ *be the index vector obtained by* $\mathcal{E}_3$ *in the first step and denote*

$$(\mathfrak{R}_0, \mathfrak{R}_1) := \mathsf{H}_1(I_1), (\gamma_{i,j}) := \mathsf{H}_2(I_2), I_3 := (3, \mathsf{crs}_{\mathsf{ISIS}}, x, \mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3, h_1, \ldots, h_\tau).$$

Then, conditioned on $v = 1$*, the extractor either returns* $(\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}}, \bar{\mathbf{r}}, \bar{\mathbf{u}}), \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/\hat{d}+\tau} \times \bar{\mathcal{C}}$ *which satisfies the following relations:*

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}_3 \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leq 2\mathcal{B}_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leq 2\mathcal{B}_2$$

and for $i = 1, 2, \ldots, \tau$ *we have*[22]

$$\begin{aligned} h_i = g_i &+ \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + \sigma(\mathbf{e}_j)^T \bar{\mathbf{y}}_3 - z_{3,j} \right) \\ &+ \sum_{j=1}^{\mathsf{n}} \gamma_{i,256+j} \cdot \left(\sigma(\mathbf{p}_j)^T \bar{\mathbf{s}} - \sigma(\boldsymbol{\beta}_j)^T \bar{\mathbf{u}} - m_{C,j} - \sigma(\mathbf{c}_{r,j})^T \bar{\mathbf{r}} \right) \\ &+ \gamma_{i,256+\mathsf{n}+1} \cdot \left(\sigma(\bar{\mathbf{s}})^T \bar{\mathbf{s}} - \mathcal{B}_s \right) + \gamma_{i,256+\mathsf{n}+2} \cdot \left(\sigma(\bar{\mathbf{r}})^T \bar{\mathbf{r}} - \mathcal{B}_r \right) \\ &+ \gamma_{i,256+\mathsf{n}+3} \cdot \sigma(\bar{\mathbf{u}} - \mathbf{x})^T \bar{\mathbf{u}} \in \hat{\mathcal{R}}_{\hat{q}}, \end{aligned} \tag{41}$$

or a $\mathsf{MSIS}_{n, m_1+m_2, \bar{\mathcal{B}}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$ *where* $\bar{\mathcal{B}} := 4\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$.

For the next step, we define the subextractor $\mathcal{E}_2$ which is informally responsible for extracting secret vectors $\vec{s}, \vec{r}, \vec{u}$ that satisfy (42). The rough intuition for $\mathcal{E}_2$ is as follows. Suppose $\mathcal{E}_2$ runs $\mathcal{E}_3$ which outputs a witness w which satisfies (41) (or a Module-SIS solution but then we are done). As before, since the ABDLOP commitment $(\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g)$ to the witness was sent before getting $(\gamma_{i,j})$ and the commitment itself is binding, this means that the extracted witness must be *independent* of the challenges $(\gamma_{i,j})$. Suppose the relevant part of w does not satisfy (24). This implies that with probability at most $\hat{p}_1^{-\tau}$ all τ equations in (41) hold. Formally, the analysis for $\mathcal{E}_2$ is almost identical to [LNP22, Lemma B.10], combined with Lemma 8.6, and thus we refer to the aforementioned result for more details.

Lemma 8.9. *For any* $\mathsf{crs}_{\mathsf{ISIS}}$*, the extractor* $\mathcal{E}_2$ *makes an expected number of at most* $12 + 11Q_{\mathsf{H}_{ISIS}}$ *queries to* $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$*. Also, for uniformly random* $\mathsf{crs}_{\mathsf{ISIS}}$*,* $\mathcal{E}_2$ *outputs* $v = 1$ *with probability at least*

$$\Pr[\mathsf{E}_3] - (Q_{\mathsf{H}_{ISIS}} + 1) \cdot \left(\frac{2}{|\mathcal{C}|} + \hat{p}_1^{-\hat{d}/2} + \hat{p}_1^{-\tau} \right).$$

Next, let $\mathcal{I} = (I_1, I_2, I_3, I_4, I_5)$ *be the index vector obtained by* $\mathcal{E}_2$ *in the first step and denote*

$$(\mathfrak{R}_0, \mathfrak{R}_1) := \mathsf{H}_1(I_1), \quad I_2 := (2, \mathsf{crs}_{\mathsf{ISIS}}, x, \mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3).$$

Then, conditioned on $v = 1$*, the extractor either returns* $(\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}}, \bar{\mathbf{r}}, \bar{\mathbf{u}}), \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/\hat{d}+\tau} \times \bar{\mathcal{C}}$ *which satisfies the following relations:*

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}_3 \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leq 2\mathcal{B}_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leq 2\mathcal{B}_2$$

[22] See the description of Equation 17 for more details.

Black-box access to: $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$

1. Run $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness $\rho \leftarrow R$ as follows: relay the $Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4$ random oracle queries to the random oracles and record all query-response pairs. Obtain $(\mathcal{I}, y, v)$. Set $i = I_2$ and let $(\gamma_{i,j})$ be the response to query i.
2. If $v = 0$, abort and return $v = 0$. Else, $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$.
3. Repeat:
   (a) sample $(\gamma'_{i,j}) \leftarrow \mathbb{Z}_{\hat{q}}^{\tau \times (256+d+3)}$,
   (b) run $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows to obtain $(\mathcal{I}', y', v')$, aborting right after (or during) the initial run of $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ if one of the following conditions holds:
   – $I'_2 \neq I_2$,
   – the input for some j-th issuing query to $\mathcal{P}^*$ is of the form $(\mathbf{m}'_j, \mathbf{r}'_j)$ where $\vec{m} = \mathsf{Coeffs}(\mathbf{m}'_j)_{\mathsf{aux}}$,
   answer the query to i with $(\gamma'_{i,j})$ while answering all the other queries consistently if the query was performed by $\mathcal{E}_4$ in the previous run, and with fresh random value otherwise,
   until a challenge $(\gamma'_{i,j})$ with $v' = 1$ and $I'_2 = I_2$ has been found.
4. If y (resp. y') is a MSIS solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathcal{I}$, y (resp. y') and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}_3, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}'_3, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathcal{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the MSIS solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. Parse $(\mathfrak{R}_0, \mathfrak{R}_1) := \mathsf{H}_1(I_1)$ and $I_2 := (2, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, \mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w}, \vec{z}_3)$.
8. Parse $\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}}, \bar{\mathbf{r}}, \bar{\mathbf{u}})$ and define $\vec{s}_1 := \mathsf{Coeffs}(\bar{\mathbf{s}}_1), \vec{s} := \mathsf{Coeffs}(\bar{\mathbf{s}}), \vec{r} := \mathsf{Coeffs}(\bar{\mathbf{r}}), \vec{u} := \mathsf{Coeffs}(\bar{\mathbf{u}})$
9. If all the following equations hold over $\mathbb{Z}_{\hat{q}}$:

$$\begin{aligned} \vec{z}_3 &= \vec{y}_3 + (\mathfrak{R}_0 - \mathfrak{R}_1)\vec{s}_1, \\ P\vec{s} &= B\vec{u} + C \begin{bmatrix} \vec{m} \\ \vec{r} \end{bmatrix}, \\ \langle \vec{s}, \vec{s} \rangle &= \mathcal{B}_s^2, \\ \langle \vec{r}, \vec{r} \rangle &= \mathcal{B}_r^2, \\ \langle \vec{u}, \vec{u} - \vec{1} \rangle &= 0, \end{aligned}$$

   return $\mathcal{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
10. Otherwise, return $v = 0$.

Fig. 24: Subextractor $\mathcal{E}_2(\mathsf{crs}_{\mathsf{ISIS}})$ as a $(Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4)$-query random oracle algorithm.

and

$$\begin{cases} \vec{z}_3 = \vec{y}_3 + (\mathfrak{R}_0 - \mathfrak{R}_1)\vec{s}_1 \\ P\vec{s} = B\vec{u} + C \begin{bmatrix} \vec{m} \\ \vec{r} \end{bmatrix} \\ \langle \vec{s}, \vec{s} \rangle = \mathcal{B}_s^2 \\ \langle \vec{r}, \vec{r} \rangle = \mathcal{B}_r^2 \\ \langle \vec{u}, \vec{u} - \vec{1} \rangle = 0 \end{cases} \tag{42}$$

where

$$\vec{s}_1 := \mathsf{Coeffs}(\bar{\mathbf{s}}_1), \quad \vec{s} := \mathsf{Coeffs}(\bar{\mathbf{s}}), \quad \vec{r} := \mathsf{Coeffs}(\bar{\mathbf{r}}), \quad \vec{u} := \mathsf{Coeffs}(\bar{\mathbf{u}}),$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{\mathcal{B}}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$ *where* $\bar{\mathcal{B}} := 4\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$.

Finally, we define the extractor $\mathcal{E}_1$ in Figure 25 which is informally responsible for extracting from the approximate range proof (c.f. Section 5.1). Intuitively, suppose $\mathcal{E}_1$ runs the subextractor $\mathcal{E}_2$ which outputs a witness $(\vec{s}_1 = (\vec{s}, \vec{r}, \vec{u}), \vec{y}_3)$ that satisfies (42) (or a Module-SIS solution), and in particular

$$\vec{z}_3 = \vec{y}_3 + \mathfrak{R}\vec{s}_1 \quad \text{and} \quad \|\vec{z}\| \leq \mathcal{B}_3$$

where $\mathfrak{R} := \mathfrak{R}_0 - \mathfrak{R}_1$. Identically as in the previous case, since the ABDLOP commitment $(\mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g)$ to the witness was sent before getting $(\mathfrak{R}_0, \mathfrak{R}_1)$ and the commitment is binding, the extracted witness must be independent of the challenges $(\mathfrak{R}_0, \mathfrak{R}_1)$. Now, suppose that $\|\vec{s}_1\| \geq \frac{\sqrt{2}}{\omega_{\min}(\lambda)} \cdot \mathcal{B}_3$. Then, by Lemma 5.4 only with negligible probability at most $2^{-\lambda}$ we can have

$$\|\vec{y}_3 + \mathfrak{R}\vec{s}_1 \bmod \hat{q}\| \leq \mathcal{B}_3.$$

In order to use the lemma, we have the requirement $\hat{q} > 16m_1\hat{d}\mathcal{B}_3$. Thus, with overwhelming probability we must have $\|\vec{s}_1\| \leq \frac{\sqrt{2}}{\omega_{\min}(\lambda)} \cdot \mathcal{B}_3$. If $\mathcal{B}_3$ is relatively small compared to the proof system modulus $\hat{q}$, then by combining with (40) we deduce the norm bounds on $\vec{s}, \vec{r}$ and that $\vec{u}$ has binary coefficients. For example, if $\mathcal{B}_3 < \frac{\omega_{\min}(\lambda)}{\sqrt{2}} \cdot \sqrt{\hat{q}}$ then

$$-\hat{q} < -\mathcal{B}_s^2 \leq \|\vec{s}\|^2 - \mathcal{B}_s^2 \leq \|\vec{s}_1\|^2 - \mathcal{B}_s^2 \leq \frac{2}{\omega_{\min}(\lambda)^2} \cdot \mathcal{B}_3^2 < \hat{q}$$

so we deduce that $\|\vec{s}\|^2 = \mathcal{B}_s^2$ over integers (and similarly for $\vec{r}$ and $\vec{u}$).

Formally, the analysis for $\mathcal{E}_1$ is almost identical to [LNP22, Lemma B.11], combined with Lemma 8.6, so we skip the proof.

Lemma 8.10. *For any* $\mathsf{crs}_{\mathsf{ISIS}}$*, the extractor* $\mathcal{E}_1(\mathsf{crs}_{\mathsf{ISIS}})$ *makes an expected number of at most* $24 + 23Q_{\mathsf{H}_{ISIS}}$ *queries to* $\mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$*. Also, for uniformly random* $\mathsf{crs}_{\mathsf{ISIS}}$*,* $\mathcal{E}_1$ *outputs* $v = 1$ *with probability at least*

$$\delta := \Pr[\mathsf{E}_3] - (Q_{\mathsf{H}_{ISIS}} + 1) \cdot \left(\frac{2}{|\mathcal{C}|} + \hat{p}_1^{-\hat{d}/2} + \hat{p}_1^{-\tau} + 2^{-\lambda} \right).$$

Next, let $\mathcal{I} = (I_1, I_2, I_3, I_4, I_5)$ *be the index vector obtained by* $\mathcal{E}_2$ *in the first step and denote* $I_1 := (1, \mathsf{crs}_{\mathsf{ISIS}}, \mathsf{x}, \mathbf{t}_A, \mathbf{t}_y, \mathbf{t}_g, \mathbf{w})$*. Then, conditioned on* $v = 1$*, the extractor either returns* $(\bar{\mathbf{s}}_1 := (\bar{\mathbf{s}}, \bar{\mathbf{r}}, \bar{\mathbf{u}}), \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/\hat{d}+\tau} \times \bar{\mathcal{C}}$ *which satisfies the following relations:*

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_y \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}_y \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}_3 \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leq 2\mathcal{B}_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leq 2\mathcal{B}_2$$

Black-box access to: $\mathcal{E}_2(\mathsf{crs}_{\mathsf{ISIS}})$

1. Run $\mathcal{E}_2(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness $\rho \leftarrow R$ as follows: relay the $Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4$ random oracle queries to the random oracles and record all query-response pairs. Obtain $(\mathcal{I}, y, v)$. Set $i = I_1$ and let $(\mathfrak{R}_0, \mathfrak{R}_1)$ be the response to query i.
2. If $v = 0$, abort and return $v = 0$. Else, $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$.
3. Repeat:
   (a) sample $(\mathfrak{R}'_0, \mathfrak{R}'_1) \leftarrow \{0,1\}^{256 \times m_1 \hat{d}} \times \{0,1\}^{256 \times m_1 \hat{d}}$,
   (b) run $\mathcal{E}_2(\mathsf{crs}_{\mathsf{ISIS}})$ with randomness ρ as follows to obtain $(\mathcal{I}', y', v')$, aborting right after (or during) the initial run of $\mathcal{P}^*$ if one of the following conditions holds:
   – $I'_1 \neq I_1$,
   – the input for some j-th issuing query to $\mathcal{P}^*$ is of the form $(\mathbf{m}'_j, \mathbf{r}'_j)$ where $\vec{m} = \mathsf{Coeffs}(\mathbf{m}'_j)_{\mathsf{aux}}$, answer the query to i with $(\mathfrak{R}'_0, \mathfrak{R}'_1)$ while answering all the other queries consistently if the query was performed by $\mathcal{E}_4$ in the previous run, and with fresh random value otherwise,
   until a challenge $(\mathfrak{R}'_0, \mathfrak{R}'_1)$ with $v' = 1$ and $I'_1 = I_1$ has been found.
4. If y (resp. y') is a MSIS solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathcal{I}$, y (resp. y') and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}_3, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}'_3, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathcal{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the MSIS solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. Define $\vec{s}_1 := \mathsf{Coeffs}(\bar{\mathbf{s}}_1)$
8. If $\|\vec{s}_1\| \leq \frac{\sqrt{2}}{\omega_{\min}(\lambda)} \cdot \mathcal{B}_3$, return $\mathcal{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
9. Otherwise, return $v = 0$.

Fig. 25: Subextractor $\mathcal{E}_1(\mathsf{crs}_{\mathsf{ISIS}})$ as a $(Q_{\mathsf{H}_{\mathsf{ISIS}}} + 4)$-query random oracle algorithm.

and

$$\begin{cases} \|\vec{s}_1\| \geq \frac{\sqrt{2}}{\omega_{\min}(\lambda)} \cdot \mathcal{B}_3 \\ P\vec{s} = B\vec{u} + C \begin{bmatrix} \vec{m} \\ \vec{r} \end{bmatrix} \\ \langle \vec{s}, \vec{s} \rangle = \mathcal{B}_s^2 \\ \langle \vec{r}, \vec{r} \rangle = \mathcal{B}_r^2 \\ \langle \vec{u}, \vec{u} - \vec{1} \rangle = 0 \end{cases} \tag{43}$$

where

$$\vec{s}_1 := \mathsf{Coeffs}(\bar{\mathbf{s}}_1), \quad \vec{s} := \mathsf{Coeffs}(\bar{\mathbf{s}}), \quad \vec{r} := \mathsf{Coeffs}(\bar{\mathbf{r}}), \quad \vec{u} := \mathsf{Coeffs}(\bar{\mathbf{u}}),$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{\mathcal{B}}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$ *where* $\bar{\mathcal{B}} := 4\nu\sqrt{\mathcal{B}_1^2 + \mathcal{B}_2^2}$.

For the next observation, we define the following events parameterised by $\mathsf{crs}_{\mathsf{ISIS}}$:

$$\mathsf{Accept}_{\mathsf{crs}_{\mathsf{ISIS}}} := \left[(\mathcal{I}, \mathsf{tr}, v, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}}) \wedge v = 1\right]$$

$$\mathsf{NoBadQuery}_{\mathsf{crs}_{\mathsf{ISIS}}} := \left[\begin{array}{c} (\mathcal{I}, \mathsf{tr}, v, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}}) \wedge v = 1 \\ \wedge\ \mathcal{E}_1(\mathsf{crs}_{\mathsf{ISIS}}) \text{ never responds to any issuing query} \\ \text{from } \mathcal{A} \text{ on input } (\mathbf{m}', \cdot) \text{ s.t. } \mathcal{I}(\vec{m}) = \mathsf{Coeffs}(\mathbf{m}')_{\mathcal{I}(\mathsf{aux})} \end{array}\right]$$

Lemma 8.11. *For any* $\mathsf{crs}_{\mathsf{ISIS}}$*, we have*

$$\Pr\left[\mathsf{Accept}_{\mathsf{crs}_{\mathsf{ISIS}}} \wedge \mathsf{NoBadQuery}_{\mathsf{crs}_{\mathsf{ISIS}}}\right] = \Pr\left[\mathsf{Accept}_{\mathsf{crs}_{\mathsf{ISIS}}}\right].$$

Proof. It follows in the similar vein as the proof of Lemma 8.6. Indeed, if $(\mathcal{I}, \mathsf{tr}, 1, \mathcal{Q}) \leftarrow \mathcal{P}^*(\mathsf{crs}_{\mathsf{ISIS}})$ and $(\vec{m}, \mathsf{aux}) := (\mathcal{I}(\vec{m}), \mathcal{I}(\mathsf{aux}))$ then by the conditions described with blue text for the extractors from Figures 22 to 25, the extractor $\mathcal{E}_1$ never answers to any issuing query on input of the form $(\mathbf{m}', \cdot)$ which satisfies $\vec{m} = \mathsf{Coeffs}(\mathbf{m}')_{\mathsf{aux}}$. We call such a query a *bad query*.

To illustrate our reasoning more precisely, we show that $\mathcal{E}_3(\mathsf{crs}_{\mathsf{ISIS}})$ never answers a bad query. The case for $\mathcal{E}_2$ and $\mathcal{E}_1$ follows similarly. In Step 1 of $\mathcal{E}_3$, it runs $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$. Clearly, $\mathcal{E}_4$ never responds to a bad query by construction. Now, if $\mathsf{Accept}_{\mathsf{crs}_{\mathsf{ISIS}}}$ holds then Step 2 of $\mathcal{E}_3$ passes.

Further, we consider the loop in Step 3 of $\mathcal{E}_3$. That is, the algorithm samples uniformly random challenges (μ_i') and runs $\mathcal{E}_4(\mathsf{crs}_{\mathsf{ISIS}})$ with the same random coins, but with reprogrammed random oracle on input I_3. So now, we look at $\mathcal{E}_4$. It runs again $\mathcal{P}^*$ (with the reprogrammed random oracle) but by the blue condition in Step 3(b) of $\mathcal{E}_3$, if $\mathcal{P}^*$ gets a bad query, we abort the current run of $\mathcal{E}_4$ and go back to Step 3(a) of $\mathcal{E}_3$. Otherwise, $\mathcal{P}^*$ outputs a tuple $(\mathcal{I}', \mathsf{tr}', v', \mathcal{Q}')$. By the condition in Step 3(b) of $\mathcal{E}_3$, if $I_3' \neq I_3$ then we go back to Step 3(a) of $\mathcal{E}_3$. We also abort the current execution of $\mathcal{E}_4$ if $v' = 0$ by Step 2 of $\mathcal{E}_4$.

Suppose we have not aborted the execution of $\mathcal{E}_4$ yet, so we are currently in the loop Step 3 of $\mathcal{E}_4$. Since $I_3' = I_3$, this implies that $\mathcal{I}'(\vec{m}) = \vec{m}$ and $\mathcal{I}'(\mathsf{aux}) = \mathsf{aux}$. Therefore, Step 3(c) of $\mathcal{E}_4$ makes sure that $\mathcal{P}^*$ does not answer any bad query. This reasoning applies to each iteration of the loop in Step 3 of $\mathcal{E}_3$. Hence, we conclude that $\mathcal{E}_3$ never responds to a bad query assuming $\mathsf{Accept}_{\mathsf{crs}_{\mathsf{ISIS}}}$ holds. □

Now, we are ready to define our strict polynomial-time reduction from Game_3 to $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ and MSIS. The reduction is given the instances of the aforementioned problems and hardwires them in Game_3. Namely, the $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ challenges $(a_1, \mathbf{a}_2, \mathbf{c}_0, \mathbf{c}_1)$ will be put inside the public key ipk, while the MSIS challenge matrix $[\mathbf{A}_1 \ \mathbf{A}_2]$ will be programmed in the $\mathsf{crs}_{\mathsf{ISIS}}$. Then, the reduction runs $\mathcal{E}_1(\mathsf{crs}_{\mathsf{ISIS}})$ but it aborts after $2(24 + 23Q_{\mathsf{H}_{\mathsf{ISIS}}})/\delta$ queries to $\mathcal{P}^*$. Further, when there is any issuing query on input $(\mathbf{m}, \mathbf{r})$ that $\mathcal{E}_1$ must answer, the reduction makes a preimage query $(\mathbf{m}, \mathbf{r})$ to the $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ oracle $\mathcal{O}_{\mathsf{pre}}(\mathbf{m}, \mathbf{r})$. This implies that $\mathcal{B}$ makes at most $O\left(\frac{Q_{\mathsf{Issue}}Q_{\mathsf{H}_{\mathsf{ISIS}}}}{\delta}\right)$ preimage queries and by Lemma 8.11, the reduction never makes a bad query. Since δ is non-negligible, the reduction runs in strict polynomial time. Finally, by Lemma 2.14, the probability that the reduction outputs a valid $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ or MSIS solution is at least $\delta/2$. □

8.3 Concrete Instantiation

In this section, we propose concrete parameters to instantiate the anonymous credentials from the construction above where we aim for 128-bit (classical) security. Note that one can directly obtain a blind signature by simply setting the number of attributes l to be one. Then, the anonymity and one-more unforgeability map to the blindness and one-more unforgeability properties of the blind signature respectively.

We split the instantiation procedure into three parts: (i) the core construction, (ii) the proof of knowledge $\Pi_{\mathsf{NIZK}}^{\mathsf{ISIS}}$, (iii) multi-proof extractable $\Pi_{\mathsf{NIZK}}^{\mathsf{Com}}$. As in many prior works, we measure the hardness of MSIS and MLWE by the root Hermite factor (we refer to [Esg20, Section 3.2.4] for more explanation on the methodology and the references therein) where e.g. $\delta \approx 1.0045$ corresponds to 128 bits of security. We bound the number of (random oracle and signing) queries by 2^{64}. Further, we assume hardness of the lattice problems with respect to expected polynomial time adversaries the same as for strict polynomial time algorithms (which is the case for currently the most efficient lattice attacks). In particular, this means that we do not need to apply Lemma 2.14 to the knowledge extractors which would additionally result with worse parameters.

Core construction. We propose the parameters in Table 3. Namely, we set $(d, q) = (4096, \approx 2^{34})$ where q is a prime congruent to 5 modulo 8. We provide a construction for $l = 8$ attributes. One then puts each attribute inside the hash function H_{M} which outputs vectors of length $\mathsf{h} = 512$ with coefficients between -2 and 2. For correctness, we select $\delta = 412$ (it will have a bigger meaning in setting parameters for one-more unforgeability). As for anonymity, we require $\mathsf{MLWE}_{1,\ell_r,\psi}$ to be hard. With the parameters proposed, the root Hermite factor is 1.00115 which corresponds to around more than 600 bits of security. We describe correctness and zero-knowledge of the proof systems separately below.

In the remainder, we focus on one-more unforgeability (c.f. Theorem 8.3) which is one of the most challenging tasks. To begin with, note that the reduction for one-more unforgeability runs in time $\mathsf{poly}(\lambda) \cdot 1/\varepsilon$,

parameter	value
d	4096
q	$17179861781 \approx 2^{34}$
N	2^{640}
t	640
h	512
l	8
ℓ_r	2
ℓ_m	1
m	3
ψ	2
$\mathfrak{s}$	$\approx 2^{27}$
δ	284
κ	56
α	12
M	3
$T_{\max}$	678

Table 3: Parameters for the main construction. More explanation behind these variables can be found in Table 2 and Theorem 7.3.

where ε is the advantage of the adversary. Since we set $\varepsilon = 2^{-128}$, we now need to consider 256-bit security for one-more unforgeability. To this end, let us first focus on all the distinguishing advantages between the security hybrids. With our parameters, we have

$$\frac{|Q_{\mathsf{H}_\mathsf{M}}|^2}{(2\psi+1)^{\mathsf{h}}} < 2^{-1100}$$

and thus we ignore this part. If we assume 257-bit security for CRS indistinguishability (Game_2) and 259-bit security for straight-line extractability (Game_3), then we aim for around 259-bit security for Game_3. Further, we will set the "knowledge error" term in Lemma 8.10 to be around 2^{-260}. Thus, we end up with 260-bit security for either the $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ or the MSIS problems. The latter one will be analysed together with $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ so we focus on $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$. To this end, we look at Theorem 7.3 where the bound on the number of queries is $Q := Q_{\mathsf{Issue}} \cdot Q_{\mathsf{H}_{\mathsf{ISIS}}} = 2^{128}$.

We instantiate Theorem 7.3 with parameters from Table 3. We analyse each subtracted term in the inequality. First, with our parameters the $\mathsf{MLWE}_{1,m+1,\psi}$ problem has the root Hermite factor 1.0005 which corresponds to more than 1000 bits of security. Thus, we loosely bound:

$$\frac{\ell_m + \ell_r}{6Q}\mathsf{Adv}^{\mathsf{MLWE}}_{1,m+1,\kappa}(\mathcal{B}) < 2^{-1000}.$$

For the next term, we had set $T_{\max} = 678$, so that $(1-1/M)^{T_{\max}} < 2^{-396}$. Thus, the term $2^\lambda/6$ in Theorem 7.3 can be written alternatively as $(1-1/M)^{T_{\max}}/6 < 2^{-398}$. Further, we can bound the rest of the terms with our parameters:

$$\frac{T_{\max}^2 Q}{12N} \approx 2^{-493}, \quad \frac{\ell_m}{3} \cdot \left(\frac{q}{(2\kappa+1)^{m+2}}\right)^d \approx 2^{-414},$$
$$\left(Q - \frac{2}{3}\right) T_{\max} \left(\frac{\varepsilon}{2M} + \frac{2\epsilon}{M} + 2^{-\delta+1}\right) \approx 2^{-394}.$$

Hence, the advantage of the reduction to the standard $\mathsf{NTRU\text{-}ISIS}_f$ is at least

$$2^{-260}/6Q - (2^{-1000} + 2^{-398} + 2^{-493} + 2^{-414} + 2^{-394}) \approx 2^{-390}.$$

parameter	explanation	value
$\hat{d}$	ring dimension of $\hat{\mathcal{R}}$	128
$\hat{p}_1$	the smallest prime divisor of $\hat{q}_2$	17179861781
$\hat{q}_2$	proof system modulus divisible by q	$\approx 2^{88}$
ξ	infinity norm on the challenges in $\mathcal{C}$	17
ν	parameter introduced in (13)	350
τ	number of repetitions for soundness	10
n	length of $\mathbf{t}_A$	30
m_2	length of the randomness vector $\mathbf{s}_2$	68

Table 4: Parameters for the proof system $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$.

parameter	explanation	value
$\hat{d}$	ring dimension of $\hat{\mathcal{R}}$	128
$\hat{p}_1$	the smallest prime divisor of $\hat{q}_1$	16253
$\hat{p}_2$	second prime divisor of $\hat{q}_1$	17179861781
$\hat{q}_1$	proof system modulus divisible by q	$\approx 2^{48}$
n	length of $\mathbf{t}_A$	18
m_2	length of the randomness vector $\mathbf{s}_2$	51
$\bar{p}$	prime modulus used for LHC	68527215
n_1	width of the matrix $\bar{\mathbf{A}}_1$ from LHC	28
n_2	width of the matrix $\bar{\mathbf{A}}_2$ from LHC	28
η_1	infinity norm of the encryption randomness $\bar{\mathbf{e}}_{1,j}$ in LHC	1
η_2	infinity norm of the encryption randomness $\bar{\mathbf{e}}_{1,j}$ in LHC	1

Table 5: Parameters for the multi-proof extractable $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$.

We heuristically assume that $\mathsf{NTRU\text{-}ISIS}_f$ is as hard as the standard $\mathsf{MSIS}_{1,m+2,\mathcal{B}}$ (or rather NTRU-SIS [PFH+17]). Recall that the MSIS bound is

$$\mathcal{B} = \mathfrak{s}\sqrt{(m+2)\cdot d} + \mathcal{B}_m \psi d\sqrt{(\ell_m+\ell_r)(m+2))} \approx 2^{33.83} < q.$$

As for the computational hardness, the root Hermite factor is 1.001425 which corresponds to more than 490 bits of security.

Parameters for $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$. As described above, we pick parameters so that the underlying proof system satisfies correctness, zero-knowledge (with 130-bit security) and knowledge soundness (with 260-bit security). To this end, we adapt the SAGE script provided in [LNP22, Section 6.1] to compute the proof size produced by $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$. We further optimise the proof by applying small tricks from [LNP22], such as the Dilithium-G compression [DLL+17] and reducing the number of garbage terms h_i by a factor of two, as well as the recent bimodal Gaussian technique described in [LN22, Section 3]. This allows us to moderately reduce the standard deviations related to rejection sampling.

We set the ring dimension $\hat{d}$ to be 128. Further, if we bound the number of oracle queries by 2^{64}, then the knowledge error term (excluding the random oracle query bound) in Lemma 8.10 must be smaller than $\approx 2^{-324}$. We set $\hat{p}_1 = q \approx 2^{34}$, so $\tau = 10$. As for the challenge space $\mathcal{C}$, in order to accommodate the 324 bits of security, we experimentally pick $(\xi, \nu) = (17, 350)$. This gives us a challenge space of size at least 2^{327}. As for the approximate range proofs, we computed $\omega_{\min}(192) = 1$ and $\omega_{\max}(192) = \sqrt{530}$ using the cumulative distribution function of chi-squared distribution with 256 degrees of freedom, as discussed in [GHL21]. Finally, the underlying computational assumption, i.e. MSIS, must hold with around 324-bit security. We satisfy this condition by picking $\hat{q} \approx 2^{88}$ and $n = 30$. The reason for such a large modulus, which we believe is the main bottleneck, is the requirement on $\hat{q}$ described in Theorem 8.3.

Multi-proof extractable $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$. We propose the parameters in Table 5. Recall that we need to satisfy correctness, CRS indistinguishability (with 257-bit security), zero-knowledge (with 130-bit security) and straightline extractability (with 259-bit security). Similarly as above, we apply the parameter selection strategy from [LNP22; LN22]. We additionally need to take into account the extractable linear homomorphic commitment (LHC) which is required for Katsumata transform [Kat21].

We first set parameters so that the underlying proof system [LNP22] works[23]. Then, we pick parameters related to the MLWE-based LHC exactly as in [Kat21, Lemma 3.13]. First, we need large enough prime $\bar{p}$ such that we can encrypt both $\mathbf{z}_1$ and $\mathbf{z}_2$. Hence, we set $\hat{p}$ such that

$$\max\left(\mathfrak{s}_1\sqrt{2m_1\hat{d}}, \mathfrak{s}_2\sqrt{2m_2\hat{d}}\right) \leq (\bar{p}-1)/4.$$

Then, we need to ensure that LHC provides no decryption errors. Consider the application where it is used, i.e. Theorem 6.3. To this end, we require for $i \in \{1,2\}$ (using notation from there):

$$\begin{aligned}
&\|(c\bar{\mathbf{t}}_{i,2} + \bar{\mathbf{w}}_{i,2}) - \bar{\mathbf{D}}_{i,1}\left(c\bar{\mathbf{t}}_{i,1} + \bar{\mathbf{w}}_{i,1}\right)\}\|_\infty \\
&\qquad = \left\|\bar{p}\cdot\left(\bar{\mathbf{D}}_{i,2}\bar{\mathbf{z}}_{i,1} - \bar{\mathbf{D}}_{i,1}\bar{\mathbf{z}}_2 + \bar{\mathbf{z}}_3\right) + \mathbf{z}_i\right\| \\
&\qquad \leq \bar{p}\cdot\left(\sqrt{n_i d}\eta_i\cdot\bar{\mathfrak{s}}_i\sqrt{2n_i d} + \sqrt{m_i d}\eta_i\cdot\bar{\mathfrak{s}}_i\sqrt{2m_i d} + \bar{\mathfrak{s}}_i\sqrt{2m_i d}\right) \\
&\qquad \leq \sqrt{2}\bar{p}\left(n_i d\eta_i + m_i d\eta_i + \sqrt{m_i d}\right)\bar{\mathfrak{s}}_i < q/2
\end{aligned} \tag{44}$$

and the lower-bounds on the standard deviations $\bar{\mathfrak{s}}_i$ are dictated by the correctness and zero-knowledge conditions in Section 6.2. Recall that we need the proof system modulus large enough to prove the norm bounds, i.e. it has to be $(\psi\sqrt{(\ell_m+\ell_r)d})^2\cdot O(\lambda)$. However, this condition is much milder than the one in (44). What are still unknown from the inequality above are the values of η_i and n_i. These need to be chosen such that $\mathsf{MLWE}_{m_i,m_i+n_i,\eta_i}$ is hard in order to provide both zero-knowledge and CRS indistinguishability. The latter one becomes the bottleneck since we aim for 257-bit security. We found that it is possible to instantiate the construction, and in particular satisfy all the aforementioned conditions, with the proof system modulus $q \approx 2^{48}$. Next, we pick $\eta_1 = \eta_2 = 1$ which results in setting $n_1 = n_2 = 28$ for the Module-LWE problem.

In order to keep the rejection rate small, we additionally apply the bimodal Gaussian technique from [LN22]. Note that, similarly as the randomness vector $\mathbf{s}_2$ for the ABDLOP commitment, the randomness vectors $\bar{\mathbf{e}}_{i,j}$ related to encrypting $\mathbf{s}_i$ are freshly created *every time* the prover algorithm is called[24]. Hence, leaking one bit of information on $\bar{\mathbf{e}}_{i,j}$ should not significantly decrease security of the protocol. Based on the observation above, we can apply bimodal Gaussian rejection as described in [LN22, Section 3.1]. with this modification we can still prove zero-knowledge but under a new assumption, called Extended-MLWE. Hence, by combining this observation with [LN22], we can simply set $\alpha_1, \alpha_2, \alpha_3, \bar{\alpha}, \bar{\alpha}_2$ all to be one. Thus, the repetition rate of the $\Pi^{\mathsf{Com}}_{\mathsf{NIZK}}$ is only 12.

As for straight-line extractability, we require the knowledge error term (without the random oracle query bound) in (34) to be at most 2^{-324}. To this end, we select parameters similarly as for $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$, e.g. the challenge space or the bounds for approximate range proofs.

Conclusion. Finally, we are ready to present various signature sizes in Table 6 (see $\mathsf{NTRU\text{-}ISIS}_f$.$\mathsf{AnonCreds}$). The public key ipk size is a single polynomial in $\mathcal{R}_q$ which is a public output of `NTRU.TrapGen`. All other parts of ipk can be generated from a seed. Hence, we obtain the public key of size $d\log q/2^{13} \approx 4096\cdot 34/2^{13} = 17$KB.

As for the communication complexity, the proof size of π is around 398KB, where $\approx 80\%$ of which is linked to LHC. We highlight that one can further reduce the proof overhead, which comes with extractable linear homomorphic commitments, by applying the NTRU-type construction from [Kat21, Section 3.5]. Then, as estimated in the aforementioned paper, it would reduce the overhead by around a factor of two, which

[23] We note that we additionally apply the Dilithium compression techniques [LNP22, Appendix A] which were not explicitly described in Section 6.

[24] In other words, Com in Fig. 11 is a one-time commitment [LNS21a].

scheme	signature	communication complexity	public key size
NTRU-ISIS$_f$.AnonCreds	243KB	473KB	17KB
Int-NTRU-ISIS$_f$.AnonCreds	62KB	107KB	3.5KB

Table 6: Signature, communication and public key sizes for the concrete instantiations of our anonymous credentials.

parameter	value
d	1024
$\hat{d}$	128
q	$33641 \approx 2^{15}$
$\hat{q}_1$	$\approx 2^{41}$
$\hat{q}_2$	$\approx 2^{50}$
N	2^{256}
t	256
h	128
l	8
ℓ_r	2
ℓ_m	1
ψ	2
$\mathfrak{s}$	658
$\mathcal{B}$	$2^{14.86}$

Table 7: Parameters for our alternative variant Int-NTRU-ISIS$_f$.AnonCreds.

gives roughly less than 150KB. However, security of the protocol would rely on so-called *decisional small matrix ratio* assumption, which does not yet seem to be a well-studied assumption. Since our construction itself already relies on new assumptions (such as NTRU-ISIS$_f$ or Extended-MLWE [LN22]), we stick with the more standard but less efficient instantiation of LHC. By counting the sizes of $(\mathbf{s}, x)$ sent by the signer, which is around 75KB, the overall communication size is $\approx$ 473KB. Finally, the signature size, which is the proof produced from Π_{NIZK}, is around 243KB.

Reducing to Int-NTRU-ISIS$_f$ *only.* There are a few artefacts of the security proof which make the parameters, and consequently the signature size, unappealing. The main bottleneck lies in the non-tight reduction from Int-NTRU-ISIS$_f$ to NTRU-ISIS$_f$ in Theorem 7.3. Let us start with the parameter m which is the length of $\mathbf{a}_2$. Namely, it allows us to do a game hop where we switch from uniformly random matrix $[\mathbf{c}_0^T|\mathbf{c}_1^T]$ to the one computed as $[\mathbf{c}_0^T|\mathbf{c}_1^T] = [a_1|\mathbf{a}_2^T|1]\mathbf{D}$ for a short random matrix $\mathbf{D}$. Note that if $m = 0$ then this reasoning would not go through. In other words, we cannot reduce distinguishing $[\mathbf{c}_0^T|\mathbf{c}_1^T] = [a_1|1]\mathbf{D}$ from a uniformly random one to Module-LWE. The reason is that in the first case we know the trapdoor for a_1 but not in the latter one. However, it is completely non-trivial whether the scheme becomes insecure when $m = 0$. This "artificial" problem has already appeared in the literature in the context of lattice-based group signatures [PLS18; LNPS21]. Another issue, which requires *long enough* m, is the witness indistinguishability argument of Theorem 7.3. Namely, we want to make sure that with an overwhelming probability, for a uniformly random vector $\mathbf{d}_0$ with coefficients in $[-\kappa, \kappa]$, there exists another one $\mathbf{d}_1$ from the same set such that $[a_1|\mathbf{a}_2^T|1]\mathbf{d}_0 = [a_1|\mathbf{a}_2^T|1]\mathbf{d}_1$. This requires us to pick $m > \frac{\log q}{\log(2\kappa+1)} - 2$ which has significant impact on the parameters. Not to mention the fact that one has to deal with the security loss Q from Theorem 7.3.

In order to investigate the limits of our construction, we propose an alternative scheme (which we call Int-NTRU-ISIS$_f$.AnonCreds in Table 6) where we ignore the issues pointed out above. In short, we heuristically

assume that the reduced $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ is as hard as the Module-SIS problem for the matrix $[\mathbf{A}\ \mathbf{C}]$ and the norm bound $\mathcal{B} := \sqrt{\mathcal{B}_s^2 + \mathcal{B}_m^2}$. In particular, we can set $m = 0$.

We set parameters following the guidelines from the previous instantiation, and we summarise our selection in Table 7. For convenience, we aim for only 124-bit security, which allows us to pick smaller ring dimension $d = 1024$. In this case, we need to ensure 254-bit security for the $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$, i.e. the MSIS problem. Then, we need to make sure the "knowledge error" term (c.f. Lemmas 6.5 and 8.10) is at most 2^{-318} to accommodate for the factor of $Q_{\mathsf{H}_{\mathsf{ISIS}}} = 2^{64}$, which comes from the Fiat-Shamir transform.

Expected polynomial-time Multi-Extract. Unfortunately, even in our milder variant we still need to account for the security loss depending on the adversary's advantage ε, since the reduction sub-algorithm, i.e. $\mathsf{Multi\text{-}Extract}$ runs in time $\mathsf{poly}(\lambda)/\varepsilon$. Because we already assume hardness of Module-SIS against expected PPT adversaries, it makes sense to make $\mathsf{Multi\text{-}Extract}$ expected polynomial time as well. Namely, in Figure 15, we allow the main loop to run until it either outputs the correct witness w or the weak opening, or it runs out of challenges to pick. The key observation is that the $\mathsf{Multi\text{-}Extract}$ algorithm is *only* run by the reduction if the initial proof is valid. Hence, by the heavy rows argument [Dam10], or its more fine-grained version in [AFK22, Lemma 4], we deduce that the expected runtime of $\mathsf{Multi\text{-}Extract}$ is $\mathsf{poly}(\lambda)$ and does not depend on ε anymore. In particular, we can reduce the bit security by 128.

For fair comparison with certain related works, e.g. [AKSY22; BLNS23], we ignore the Fiat-Shamir transform loss $Q_{\mathsf{H}_{\mathsf{ISIS}}}$ which was also not explicitly considered in the aforementioned constructions. Thus, the signature is a proof size for $\Pi^{\mathsf{ISIS}}_{\mathsf{NIZK}}$ with ≈ 128-bit security. In Table 1 we provide the signature sizes for both cases: either reducing to $\mathsf{NTRU\text{-}ISIS}_f$ or $\mathsf{Int\text{-}NTRU\text{-}ISIS}_f$ using the same methodology as described throughout this section.

Acknowledgments

This work is supported by the EU H2020 ERC Project 101002845 PLAZA.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. "Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE". In: *CRYPTO*. 2010, pp. 98–115.

[ACL+22] M. R. Albrecht, V. Cini, R. W. F. Lai, G. Malavolta, and S. A. Thyagarajan. "Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable". In: *CRYPTO*. Lecture Notes in Computer Science. 2022.

[AFK22] T. Attema, S. Fehr, and M. Klooß. "Fiat-Shamir Transformation of Multi-round Interactive Proofs". In: *TCC (1)*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 113–142.

[Ajt96] M. Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *STOC*. 1996, pp. 99–108.

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

[ALS20] T. Attema, V. Lyubashevsky, and G. Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO (2)*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 470–499.

[ASM08] M. H. Au, W. Susilo, and Y. Mu. "Constant-Size Dynamic k-TAA". In: *IACR Cryptol. ePrint Arch.* (2008), p. 136. URL: `http://eprint.iacr.org/2008/136`.

[ASY21] S. Agrawal, D. Stehlé, and A. Yadav. "Towards Practical and Round-Optimal Lattice-Based Threshold and Blind Signatures". In: *IACR Cryptol. ePrint Arch.* (2021), p. 381.

[Ban93] W. Banaszczyk. "New bounds in some transference theorems in the geometry of numbers". In: *Mathematische Annalen* 296.1 (1993), pp. 625–635. ISSN: 1432-1807. DOI: `10.1007/BF01445125`. URL: `https://doi.org/10.1007/BF01445125`.

[BBS04] D. Boneh, X. Boyen, and H. Shacham. "Short Group Signatures". In: *IACR Cryptol. ePrint Arch.* (2004), p. 174. URL: `http://eprint.iacr.org/2004/174`.

[BDE+18] J. Bootle, C. Delaplace, T. Espitau, P. Fouque, and M. Tibouchi. "LWE Without Modular Reduction and Improved Side-Channel Attacks Against BLISS". In: *ASIACRYPT (1)*. Vol. 11272. Lecture Notes in Computer Science. Springer, 2018, pp. 494–524.

[BDK+18] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy, EuroS&P*. 2018, pp. 353–367.

[BDL+18] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. "More Efficient Commitments from Structured Lattice Assumptions". In: *SCN*. 2018, pp. 368–385.

[BJRW20] K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. "Towards Classical Hardness of Module-LWE: The Linear Rank Case". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 289–317.

[BLNS23] W. Beullens, V. Lyubashevsky, N. K. Nguyen, and G. Seiler. *Lattice-Based Blind Signatures: Short, Efficient, and Round-Optimal*. Cryptology ePrint Archive, Paper 2023/077. `https://eprint.iacr.org/2023/077`. 2023. URL: `https://eprint.iacr.org/2023/077`.

[BLS19] J. Bootle, V. Lyubashevsky, and G. Seiler. "Algebraic Techniques for Short(er) Exact Lattice-Based Zero-Knowledge Proofs". In: *CRYPTO (1)*. Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, pp. 176–202.

[BMW03] M. Bellare, D. Micciancio, and B. Warinschi. "Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions". In: *EUROCRYPT*. 2003, pp. 614–629.

[BS22] W. Beullens and G. Seiler. "LaBRADOR: Compact Proofs for R1CS from Module-SIS". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1341.

[BTT22] C. Boschini, A. Takahashi, and M. Tibouchi. *MuSig-L: Lattice-Based Multi-Signature With Single-Round Online Phase*. Cryptology ePrint Archive, Paper 2022/1036. `https://eprint.iacr.org/2022/1036`. 2022. URL: `https://eprint.iacr.org/2022/1036`.

[CDL16] J. Camenisch, M. Drijvers, and A. Lehmann. "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited". In: *TRUST*. Vol. 9824. Lecture Notes in Computer Science. Springer, 2016, pp. 1–20.

[Cha85] D. Chaum. "Security Without Identification: Transaction Systems to Make Big Brother Obsolete". In: *Commun. ACM* 28.10 (1985), pp. 1030–1044. DOI: `10.1145/4372.4373`. URL: `https://doi.org/10.1145/4372.4373`.

[Che95] L. Chen. "Access with Pseudonyms". In: *Cryptography: Policy and Algorithms, International Conference, Brisbane, Queensland, Australia, July 3-5, 1995, Proceedings*. Ed. by E. Dawson and J. D. Golic. Vol. 1029. Lecture Notes in Computer Science. Springer, 1995, pp. 232–243. DOI: `10.1007/BFb0032362`. URL: `https://doi.org/10.1007/BFb0032362`.

[CKS09] J. Camenisch, M. Kohlweiss, and C. Soriente. "An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials". In: *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*. Ed. by S. Jarecki and G. Tsudik. Vol. 5443. Lecture Notes in Computer Science. Springer, 2009, pp. 481–500. DOI: `10.1007/978-3-642-00468-1\_27`. URL: `https://doi.org/10.1007/978-3-642-00468-1\_27`.

[CKS10] J. Camenisch, M. Kohlweiss, and C. Soriente. "Solving Revocation with Efficient Update of Anonymous Credentials". In: *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*. Ed. by J. A. Garay and R. D. Prisco. Vol. 6280. Lecture Notes in Computer Science. Springer, 2010, pp. 454–471. DOI: `10.1007/978-3-642-15317-4\_28`. URL: `https://doi.org/10.1007/978-3-642-15317-4\_28`.

[CL01] J. Camenisch and A. Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*. Ed. by B. Pfitzmann. Vol. 2045. Lecture Notes in Computer Science. Springer, 2001, pp. 93–118. DOI: `10.1007/3-540-44987-6\_7`. URL: `https://doi.org/10.1007/3-540-44987-6\_7`.

[CL02a] J. Camenisch and A. Lysyanskaya. "A Signature Scheme with Efficient Protocols". In: *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*. Ed. by S. Cimato, C. Galdi, and G. Persiano. Vol. 2576. Lecture Notes in Computer Science. Springer, 2002, pp. 268–289. DOI: `10.1007/3-540-36413-7\_20`. URL: `https://doi.org/10.1007/3-540-36413-7\_20`.

[CL02b] J. Camenisch and A. Lysyanskaya. "Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials". In: *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*. Ed. by M. Yung. Vol. 2442. Lecture Notes in Computer Science. Springer, 2002, pp. 61–76. DOI: 10.1007/3-540-45708-9_5. URL: https://doi.org/10.1007/3-540-45708-9_5.

[Dam10] I. Damgård. *On Σ-Protocols*. Lecture on Cryptologic Protocol Theory; Faculty of Science, University of Aarhus. 2010.

[Dam88] I. Damgård. "Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals". In: *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*. Ed. by S. Goldwasser. Vol. 403. Lecture Notes in Computer Science. Springer, 1988, pp. 328–335. DOI: 10.1007/0-387-34799-2_26. URL: https://doi.org/10.1007/0-387-34799-2_26.

[DKL+18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.1 (2018), pp. 238–268.

[DLL+17] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehle. *CRYSTALS – Dilithium: Digital Signatures from Module Lattices*. Cryptology ePrint Archive, Report 2017/633. https://ia.cr/2017/633. "The Dilithium-G" scheme can be found in the June 2017 version of this report. The direct url is https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2017/633&version=20170627:201152&file=633.pdf. 2017.

[DLP14] L. Ducas, V. Lyubashevsky, and T. Prest. "Efficient Identity-Based Encryption over NTRU Lattices". In: *ASIACRYPT*. 2014, pp. 22–41.

[DN12] L. Ducas and P. Q. Nguyen. "Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures". In: *ASIACRYPT*. 2012, pp. 433–450.

[ENS20] M. F. Esgin, N. K. Nguyen, and G. Seiler. "Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings". In: *ASIACRYPT (2)*. 2020, pp. 259–288.

[Esg20] M. F. Esgin. "Practice-Oriented Techniques in Lattice-Based Cryptography". In: (2020). DOI: 10.26180/5eb8f525b3562. URL: https://bridges.monash.edu/articles/thesis/Practice-Oriented_Techniques_in_Lattice-Based_Cryptography/12279728.

[ESLL19] M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. "Lattice-Based Zero-Knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications". In: *CRYPTO (1)*. Springer, 2019, pp. 115–146.

[EZS+19] M. F. Esgin, R. K. Zhao, R. Steinfeld, J. K. Liu, and D. Liu. "MatRiCT: Efficient, Scalable and Post-Quantum Blockchain Confidential Transactions Protocol". In: *CCS*. ACM, 2019, pp. 567–584.

[Fis06] M. Fischlin. "Round-Optimal Composable Blind Signatures in the Common Reference String Model". In: *CRYPTO*. Vol. 4117. Lecture Notes in Computer Science. Springer, 2006, pp. 60–77.

[FS86] A. Fiat and A. Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO*. 1986, pp. 186–194.

[GHL21] C. Gentry, S. Halevi, and V. Lyubashevsky. "Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties". In: *IACR Cryptol. ePrint Arch.* (2021), p. 1397.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *STOC*. 2008, pp. 197–206.

[HM17] G. Herold and A. May. "LP Solutions of Vectorial Integer Subset Sums - Cryptanalysis of Galbraith's Binary Matrix LWE". In: *Public Key Cryptography (1)*. Vol. 10174. Lecture Notes in Computer Science. Springer, 2017, pp. 3–15.

[Hypa] Hyperledger Foundation. *Hyperledger Aries*. https://www.hyperledger.org/use/aries. [Online; accessed 06-October-2022].

[Hypb] Hyperledger Foundation. *Hyperledger Indy*. https://www.hyperledger.org/use/hyperledger-indy. [Online; accessed 06-October-2022].

[JRLS22] C. Jeudy, A. Roux-Langlois, and O. Sanders. *Lattice Signature with Efficient Protocols, Application to Anonymous Credentials*. Cryptology ePrint Archive, Paper 2022/509. https://eprint.iacr.org/2022/509. 2022. URL: https://eprint.iacr.org/2022/509.

[Kat21] S. Katsumata. "A New Simple Technique to Bootstrap Various Lattice Zero-Knowledge Proofs to QROM Secure NIZKs". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 580–610.

[LKWL22] T. Looker, V. Kalos, A. Whitehead, and M. Lodder. *The BBS Signature Scheme*. https://www.ietf.org/id/draft-looker-cfrg-bbs-signatures-01.html. [Online; accessed 06-October-2022]. 2022.

[LM13] V. Lyubashevsky and D. Micciancio. *Asymptotically Efficient Lattice-Based Digital Signatures*. Cryptology ePrint Archive, Paper 2013/746. https://eprint.iacr.org/2013/746. 2013. URL: https://eprint.iacr.org/2013/746.

[LM18] V. Lyubashevsky and D. Micciancio. "Asymptotically Efficient Lattice-Based Digital Signatures". In: *J. Cryptol.* 31.3 (2018). Preliminary version appeared in TCC 2008, pp. 774–797. URL: https://eprint.iacr.org/2013/746.

[LMPY16] B. Libert, F. Mouhartem, T. Peters, and M. Yung. "Practical "Signatures with Efficient Protocols" from Simple Assumptions". In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016, Xi'an, China, May 30 - June 3, 2016*. Ed. by X. Chen, X. Wang, and X. Huang. ACM, 2016, pp. 511–522. DOI: 10.1145/2897845.2897898. URL: https://doi.org/10.1145/2897845.2897898.

[LN22] V. Lyubashevsky and N. K. Nguyen. *BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications*. Cryptology ePrint Archive, Paper 2022/1307. https://eprint.iacr.org/2022/1307. 2022. URL: https://eprint.iacr.org/2022/1307.

[LNP22] V. Lyubashevsky, N. K. Nguyen, and M. Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *IACR Cryptol. ePrint Arch.* (2022). Appears in Crypto 2022, p. 284.

[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 (4)*. Springer, 2021, pp. 218–248.

[LNS21a] V. Lyubashevsky, N. K. Nguyen, and G. Seiler. "Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments". In: *Public Key Cryptography (1)*. Springer, 2021, pp. 215–241.

[LNS21b] V. Lyubashevsky, N. K. Nguyen, and G. Seiler. "SMILE: Set Membership from Ideal Lattices with Applications to Ring Signatures and Confidential Transactions". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 611–640.

[LRSW99] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. "Pseudonym Systems". In: *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings*. Ed. by H. M. Heys and C. M. Adams. Vol. 1758. Lecture Notes in Computer Science. Springer, 1999, pp. 184–199. DOI: 10.1007/3-540-46513-8_14. URL: https://doi.org/10.1007/3-540-46513-8_14.

[LS15] A. Langlois and D. Stehlé. "Worst-case to average-case reductions for module lattices". In: *Designs, Codes and Cryptography* 75.3 (2015), pp. 565–599.

[Lyu09] V. Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures". In: *ASIACRYPT*. 2009, pp. 598–616.

[Lyu12] V. Lyubashevsky. "Lattice Signatures Without Trapdoors". In: *EUROCRYPT*. 2012, pp. 738–755.

[MATa] MATTR. *bbs-signatures*. https://github.com/mattrglobal/bbs-signatures. [Online; accessed 06-October-2022].

[MATb] MATTR. *MATTR*. https://github.com/mattrglobal. [Online; accessed 06-October-2022].

[MR07] D. Micciancio and O. Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". In: *SIAM J. Comput.* 37.1 (2007), pp. 267–302.

[NFC22] NFCW. *Digital identity market revenues to reach US$53bn in 2026*. https://www.nfcw.com/2022/01/31/375825/digital-identity-market-revenues-to-reach-us53bn-in-2026/. [Online; accessed 06-October-2022]. 2022.

[PFH+17] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *FALCON*. Tech. rep. https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions. National Institute of Standards and Technology, 2017.

[PK22] R. del Pino and S. Katsumata. *A New Framework For More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling*. Cryptology ePrint Archive, Paper 2022/834. https://eprint.iacr.org/2022/834. 2022. URL: https://eprint.iacr.org/2022/834.

[PLS18] R. del Pino, V. Lyubashevsky, and G. Seiler. "Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability". In: *ACM Conference on Computer and Communications Security*. ACM, 2018, pp. 574–591.

[Reg09] O. Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (2009).

[The22] The AnonCreds Specification Working Group. *The AnonCreds Specification*. https://github.com/AnonCreds-WG/anoncreds-spec. [Online; accessed 06-October-2022]. 2022.

[Ver] Veramo. *Veramo core development*. `https://github.com/uport-project`. [Online; accessed 06-October-2022].

[WW22] H. Wee and D. J. Wu. *Succinct Vector, Polynomial, and Functional Commitments from Lattices*. Cryptology ePrint Archive, Paper 2022/1515. `https://eprint.iacr.org/2022/1515`. 2022. URL: `https://eprint.iacr.org/2022/1515`.

[YAZ+19] R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. "Efficient Lattice-Based Zero-Knowledge Arguments with Standard Soundness: Construction and Applications". In: *CRYPTO (1)*. Springer, 2019, pp. 147–175.