Relaxed Lattice-Based Signatures with Short Zero-Knowledge Proofs

Cecilia Boschini, Jan Camenisch, and Gregory Neven

IBM Research – Zurich
{bos, jca, nev}@zurich.ibm.com

Abstract. Higher-level cryptographic privacy-enhancing protocols such as anonymous credentials, voting schemes, and e-cash are often constructed by suitably combining signature, commitment, and encryption schemes with zero-knowledge proofs. Indeed, a large body of protocols have been constructed in that manner from Camenisch-Lysyanskaya signatures and generalized Schnorr proofs. In this paper, we build a similar framework for lattice-based schemes by presenting a signature and commitment scheme that are compatible with Lyubashevsky's Fiat-Shamir proofs with abort, currently the most efficient zero-knowledge proofs for lattices. To cope with the relaxed soundness guarantees of these proofs, we define corresponding notions of relaxed signature and commitment schemes. We demonstrate the flexibility and efficiency of our new primitives by constructing a new lattice-based anonymous attribute token scheme and providing concrete parameters to securely instantiate this scheme.

1 Introduction

An established and successful way to construct privacy-enhancing cryptographic protocols is to suitably combine various primitives such as signatures, commitments, and encryption schemes with efficient zero-knowledge proofs. Examples of such constructions include blind signatures [AO09,Fis06], group signatures [BMW03,KY05], direct anonymous attestation [BCC04], electronic cash [CFN90], voting schemes [HS00], adaptive oblivious transfer [CNs07,CDNZ11], and anonymous credentials [BCKL08,CL01].

One of the crucial building blocks is a signature scheme with efficient zero-knowledge proofs of knowledge of a signature on a hidden message. Commitment schemes are also common ingredients, either as "glue" to bridge zero-knowledge proofs over different cryptographic primitives [CKL$^+$16], or to facilitate zero-knowledge proofs by hiding the message or certain components of the signature [ACJT00,CL03,BBS04].

One can of course use generic zero-knowledge techniques [GMW86] to combine cryptographic schemes, but to get truly efficient constructions, one needs schemes that interact well with each other and allow for efficient zero-knowledge proofs. A well known set of such schemes consist of Camenisch-Lysyanskaya signatures [CL03], Damgård-Fujisaki commitments [DF02], and Camenisch-Shoup

verifiable encryption [CS03]. They can be combined using generalized Schnorr proofs [CKY09] and the Fiat-Shamir transform [FS87] into efficient proofs of relations between their (committed) inputs and (committed) outputs. More recently, an alternative set of primitives has emerged, so-called structure preserving primitives [AFG+10,CHK+11], that use Groth-Sahai proofs [GS08] as a framework to create zero-knowledge proofs.

All of the above schemes, however, are based on hardness assumptions related to factoring large integers and computing discrete logarithms, which are known to succumb to attacks on quantum computers. To guarantee security on the long term, it would be best to switch to quantum-resistant problems such as lattices. Indeed, a number of cryptographic primitives whose security relies on lattice-based assumptions have been proposed. While several lattice-based schemes exist for basic tasks such as signatures and encryption, these schemes usually do not lend themselves very well to efficient zero-knowledge proofs.

Most lattice-based zero-knowledge proofs are either Fiat-Shamir proofs with single-bit challenges or Stern-type proofs [Ste94]. Because of the large soundness errors of 1/2 and 2/3 that these proofs incur, respectively, they have to be repeated many times in parallel, which comes at a considerable cost in efficiency. Lyubashevsky's "Fiat-Shamir with Aborts" technique [Lyu12] yields much more efficient proofs with large challenges, but these proofs have the disadvantage that they are "relaxed", in the sense that extracted witnesses are only guaranteed to lie in a considerably larger domain than the witnesses used to construct the proof.

1.1 Our Results

In this paper, we provide a signature and a commitment scheme with efficient zero-knowledge proofs using Lyubashevsky's Fiat-Shamir with aborts technique. To be compatible with the "relaxed" extraction of such zero-knowledge proofs, we define "relaxed" signature and commitment schemes, in the sense that the verification algorithms accept messages, signatures, and openings that are never output by the honest signing or committing algorithms. By allowing exactly the relaxation induced by the extraction of zero-knowledge proofs, and by proving that our schemes remain secure under a suitably adapted notion in spite of that relaxation, we obtain efficient and securely composable zero-knowledge proofs for lattice-based primitives.

We demonstrate the use of our signature and commitment schemes in the construction of privacy-enhancing technologies by building an anonymous attribute token (AAT) scheme [CNR12]. An AAT scheme enables users to obtain credentials with multiple attributes, so that they can selectively and disclose these attributes to a verifiers in an unlinkable fashion.

We suggest concrete parameter choices for our schemes that yield a secure yet efficient instantiation. We follow the approach of Alkim et al. [ADPS16] and present different sets of parameters, ranging from conservative, quantum-resistant choices to more liberal estimates that only guarantee classical security. Even in our most conservative analysis, assuming the hardness of Ring-SIS and

Ring-LWE through a complexity leveraging argument, we obtain presentation token sizes less than 15 MB, which is well below the signature sizes or related lattice-based primitives [LLNW16]. In our least conservative analysis, assuming the hardness of two new interactive assumptions, we even obtain presentation tokens as small as 1.5 MB, which can be considered for practical use.

Finally, we explore the flexibility of our primitives by combining our relaxed signature and commitment schemes with the relaxed verifiable encryption scheme by Lyubashevsky and Neven [LN17] to obtain AATs with opening, which can be easily modified to become a group signature scheme (see Appendix G). The resulting scheme cannot be considered efficient though. We only included it as a proof of concept. Analogously to the non-lattice-based world where generic, modular constructions [CKL+16] are often considerably less efficient than direct schemes [ACJT00,CL03], we expect that a more efficient direct construction could potentially be built by breaking open the different building blocks. This is left as an open problem, as the main focus of this paper is to study a new set of compatible, lattice-based primitives for the design of privacy-preserving protocols.

1.2 Related Work

The only known lattice-based anonymous attribute token scheme [CNR12] has presentation token sizes that are linear in the number of group members, and is therefore mainly a proof of concept. Our AAT scheme is the first that could be considered suitable for practical applications in a post-quantum world[1].

Our proposal of lattice-based signature with protocols is not the first attempt to design efficient cryptographic building blocks. In a concurrent work, Libert et al. [LLM+16] presented a signature scheme with proofs based on a Stern-type ZK protocol. Moreover, there exists a line of work on lattice-based group signatures that combines signature schemes (usually variants of Boyen's signature [Boy10] or Böhl signature [BHJ+15]) with non-interactive zero-knowledge (NIZK) protocols, usually either Stern-type NIZK protocols (cfr. [LLNW14,LNW15,LLM+16]), or Lyubashevsky proofs [Lyu12] with single-bit challenges (cfr. [LLLS13,NZZ15])[2]. The advantage of using these protocols is that it is possible to prove knowledge of a witness for the exact relation, thus it is not necessary to relax the verification algorithms. The drawback is that Stern-type protocols have soundness error of $2/3$ and Lyubashevsky proofs with single-bit challenges of $1/2$, thus they require to be repeated a number of times that is linear in the security parameter to have a negligible soundness error. This reflects in parameters choices and sizes: as it was already observed by Lib-

[1] We do not claim ours to be the first practical AAT. In fact, an AAT scheme based on discrete log is at the core of Microsoft's U-Prove [PZ].

[2] We do not consider in our comparison the lattice-based group signature built by Benhamouda et al. [BCK+14]. Indeed, it is a special case, as the authors avoided expensive zero-knowledge proofs on lattice signatures by bridging a lattice-based encryption scheme to a non-lattice-based signature scheme.

ert et al. [LLNW16], all the schemes proposed until now output signature of size greater than 61 MB.

2 Preliminaries

2.1 Notation

If A is a probabilistic algorithm, then by $\mathsf{A}(x)$ we denote the output distribution of A on input x and run with uniformly chosen random coins. Computing y with A on input x amounts to choose y from the distribution $\mathsf{A}(x)$, denoted by $y \xleftarrow{\$} \mathsf{A}(x)$. We write $y \in \mathsf{A}(x)$ if the probability that $\mathsf{A}(x)$ will output y is non-zero. We use $\mathsf{A}^{\mathcal{H}}$ to denote the fact that A has oracle access to the function $\mathcal{H}$. A function $\nu(n)$ is said to be *negligible* if $\nu(n) \leq \frac{1}{p(n)}$ for any polynomial $p(n)$ and sufficiently large n. Throughout the paper we denote by λ the security parameter of a scheme.

Let L be a NP language. We associate with any L a polynomial-time recognizable relation R_L that defines L itself: $L = \{x : \exists w \text{s.t. } (x, w) \in R_L\}$, where w is called a *witness* for the instance x.

2.2 Polynomial Rings

Let $\mathcal{R}_q = \mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ be a polynomial ring for a prime q. Operations are the usual addition and multiplication modulo q and $\mathbf{x}^n + 1$. An element of $\mathcal{R}_q$ is a polynomial $\mathbf{a} = \sum_{i=0}^{n-1} a_i \mathbf{x}^i$, where $a_i \in \{-(q-1)/2, \ldots, (q-1)/2\}$. A matrix in $\mathcal{R}_q^{n \times m}$ will be denoted by bold upper-case letters. We define the following norms on the set of polynomials: $\|\mathbf{a}\|_1 = \sum_{i=0}^{n-1} |a_i|$, $\|\mathbf{a}\|_\infty = \max_i |a_i|$ and $\|\mathbf{a}\| = \sqrt{\sum_{i=0}^{n-1} a_i^2}$. A *small* element of the ring will be a polynomial in $\mathcal{R}_q$ with small coefficient w.r.t. one of these norms depending on the context.

With $\mathbf{b} \leftarrow \mathcal{R}_q$ we will mean that the polynomial $\mathbf{b}$ is sampled uniformly at random from $\mathcal{R}_q$. For two matrices $\mathbf{A}$ and $\mathbf{B}$, we will denote by $[\mathbf{A}|\mathbf{B}]$ their horizontal concatenation and with $[\mathbf{A}; \mathbf{B}]$ their vertical concatenation. We denote row vectors by $[\mathbf{a}\ \mathbf{b}]$ and column vectors as $[\mathbf{a}; \mathbf{b}]$. With $\mathbf{1}_m$ we will indicate the vector of length m whose components are equal to $\mathbf{1}$, $\mathbf{0}_{m_1 \times m_2}$ (resp., $\mathbf{0}_m$) will be the zero matrix (resp., vector) of dimension $m_1 \times m_2$ (resp., m) and $\mathbb{I}_m$ the identity matrix of dimension m. The norms of a vector $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_k]$ are defined as $\|\mathbf{V}\|_\infty = \max_i \|\mathbf{v}_i\|_\infty$ and $\|\mathbf{V}\| = \sqrt{\sum_i \|\mathbf{v}_i\|^2}$. With $\mathcal{R}_3$ we denote the ring of polynomials with coefficients in $\mathbb{Z}_3 = \{0, \pm 1\}$. Throughout the paper we will consider these element as also element of the subset of $\mathcal{R}_q$ of polynomials with coefficients in $\{\pm 1, 0\}$ using a standard mapping.

The ring $\mathcal{R}_q$ has some very useful properties. First, for any $K|n$ and integer p, we can construct a subring of $\mathcal{R}_q$ as the subset of elements $\mathbf{a} \in \mathcal{R}_q$ such that $\mathbf{a} = a_0 + a_1 \mathbf{x}^{n/K} + a_2 \mathbf{x}^{2n/K} + \ldots + a_{K-1} \mathbf{x}^{(K-1)n/K}$ and $a_i \in \{-(p-1)/2, \ldots, (p-1)/2\}$. Such subring will be denoted by $\mathcal{R}_p^{(K)}$. Observe that $\mathcal{R}_p^{(K)}$ is isomorphic to $\mathbb{Z}_p[x]/\langle \mathbf{x}^K + 1\rangle$.

Among others, the choice of q strongly influences the number of invertible elements that can be found in the ring. Following the approach by Lyubashevsky et al. ([LN17]), we set q to be such that $q \equiv 5 \bmod 8$, so that all the elements with small enough coefficient are guaranteed to be invertible.

Lemma 1 ([LN17], Lemma 2.2). *Let $\mathcal{R}_q = \mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$ where $n > 1$ is a power of 2 and q is a prime congruent to 5 mod 8. This ring has exactly $2q^{n/2} - 1$ elements without an inverse. Moreover, every non-zero polynomial $\mathbf{a}$ in $\mathcal{R}_q$ with $\|\mathbf{a}\|_\infty < \sqrt{q/2}$ has an inverse.*

Finally, we bound the norm of a product of polynomials in $\mathcal{R}_q$. The proof of the lemma is straightforward and it can be found in Appendix D.

Lemma 2. *Let $\mathbf{a},\ \mathbf{b} \in \mathcal{R}_q$ be such that $n\|\mathbf{a}\|_\infty \cdot \|\mathbf{b}\|_\infty \leq (q-1)/2$. Then we have that $\|\mathbf{ab}\| \leq \|\mathbf{a}\|\|\mathbf{b}\|\sqrt{n}$ and $\|\mathbf{ab}\|_\infty \leq \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty n \leq \frac{q-1}{2}$.*

We denote by $Inv(\mathcal{R}_q)$ the set of all the invertible polynomials in $\mathcal{R}_q$.

2.3 Lattices

An integer lattice is an additive subgroup of $\mathbb{Z}^n$. It is generated by a basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \in \mathbb{Z}^{n \times m}$, and m is called *dimension* of the lattice. If $k = n$ and the vectors in the basis are linearly independent the lattice is a *full-rank* lattice. The Gram-Schmidt orthogonalization of a full-rank basis $\mathbf{B}$ is $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_1, \ldots \tilde{\mathbf{b}}_n\}$ where $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\|\tilde{\mathbf{b}}_j\|} \tilde{\mathbf{b}}_j$ for the usual definition of Euclidean norm and scalar product. Given a basis $\mathbf{B}$, we write $\Lambda = \mathcal{L}(\mathbf{B})$ to indicate that the lattice Λ is generated by $\mathbf{B}$. Given a vector $\mathbf{v} \in \mathbb{Z}^n$, a *coset* $\Lambda + \mathbf{v}$ of a lattice Λ is the set $\{\mathbf{a} + \mathbf{v}\}_{\mathbf{a} \in \Lambda}$. Let $\tilde{\lambda}(\mathcal{L}(\mathbf{B})) = \min_{\mathbf{B}' \text{ s.t. } \mathcal{L}(\mathbf{B}') = \mathcal{L}(\mathbf{B})} \|\tilde{\mathbf{B}}'\|$. For a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, the lattice $\Lambda^\perp$ is the lattice: $\Lambda^\perp = \mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Ax} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^m$.

We define the *discrete Gaussian distribution* centered in $\mathbf{c}$ with standard deviation σ on a full-rank lattice Λ as $\mathcal{D}_{\Lambda,\mathbf{c},\sigma}(\mathbf{v}) = e^{-\frac{\pi\|\mathbf{v}-\mathbf{c}\|^2}{\sigma^2}} / \sum_{\mathbf{u} \in \Lambda} e^{-\frac{\pi\|\mathbf{u}-\mathbf{c}\|^2}{\sigma^2}}$ for all $\mathbf{v} \in \Lambda$, and 0 on all the other points in the space. Let $\mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}$ be the distribution of the vectors $\mathbf{s}$ such that $\mathbf{s} \sim \mathcal{D}_{\mathbb{Z}^n,\mathbf{0},\sigma}$ conditioned on $\mathbf{As} = \mathbf{u} \bmod q$. If $\sigma > \|\tilde{\mathbf{B}}\|\sqrt{\log(n)}$ we can sample from this distribution using a basis $\mathbf{B}$ of $\mathcal{L}^\perp(\mathbf{A})$ (cfr. [GPV08,BLP+13]). Vectors sampled from such distribution, have norm bounded by the following lemma.

Lemma 3 (Lemma 1.5 in [Ban93] and Lemma 4.4 in [Lyu12]). *Let $\mathbf{A} \in \mathbb{Z}^{n \times m}$ with $2^{11} < m$ and $\mathbf{u} \in \mathbb{Z}_q^n$. For $\sigma \geq \tilde{\lambda}(\mathcal{L}^\perp(\mathbf{A}))$ it holds:*
1. $\mathsf{Pr}_{\mathbf{s} \xleftarrow{\$} \mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}}(\|\mathbf{s}\| > 1.05\sigma\sqrt{m}) < 2^{-5}$.
2.[3] $\mathsf{Pr}_{\mathbf{s} \xleftarrow{\$} \mathcal{D}^\perp_{\mathbf{A},\mathbf{u},\sigma}}(\|\mathbf{s}\|_\infty > 8\sigma) < m2^{-25}$.
In particular, the inequalities hold also when $\mathbf{s} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\mathbf{u},\sigma}$.

[3] This bound is looser than normal because we are taking a σ that is only $\tilde{\lambda}(\mathcal{L}^\perp(\mathbf{A}))$. If we were to impose that $\sigma > 2\tilde{\lambda}(\mathcal{L}^\perp(\mathbf{A}))$, then the probability would be smaller.

Observe that it is enough that the bound holds with non-negligible probability, as each time we sample we can check the norm of the vector and discard it if the norm is too large.

Finally, we define the largest singular value, a quantity that is used to measure the geometric quality of a lattice basis. Given a matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ its *largest singular value* is: $\mathsf{s}_1(\mathbf{R}) = \max_{\mathbf{u} \in \mathbb{R}^m} \frac{\|\mathbf{Ru}\|}{\|\mathbf{u}\|}$. It follows from the definition that for every matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ and vector $\mathbf{u} \in \mathbb{R}^m$ it holds that $\|\mathbf{Ru}\| \leq \mathsf{s}_1(\mathbf{R})\|\mathbf{u}\|$.

2.4 Polynomial Lattices

A m-dimensional polynomial lattice is an additive subgroup of $\mathcal{R}_q$, where a basis is a vector $\mathbf{B} \in \mathcal{R}_q^{1 \times m}$. Given a vector $\mathbf{A} \in \mathcal{R}_q^{1 \times m}$ we define the m-dimensional lattice $\mathcal{L}^\perp(\mathbf{A})$ as $\Lambda^\perp = \mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{V} \in (\mathbb{Z}[\mathbf{x}]/\langle \mathbf{x}^n + 1 \rangle)^m \mid \mathbf{AV} = \mathbf{0} \bmod q\} \subseteq \mathcal{R}_q^m$. Consider the obvious embedding that maps a polynomial to the vector of its coefficients. Then $\Lambda^\perp$ can be also seen as a nm-dimensional integer lattice over $\mathbb{Z}$. To generate a discrete Gaussian sample, we can generate a sample over $\mathbb{Z}^n$ and then map it into $\mathcal{R}_q$ using the obvious embedding of coordinates into coefficients of the polynomials. With a slight abuse of notation, we will write $\mathbf{y} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q, \mathbf{u}, \sigma}$ to indicate that $\mathbf{y}$ was sampled from $\mathcal{D}_{\mathbb{Z}^n, \mathbf{u}, \sigma}$ and then mapped to $\mathcal{R}_q$. Similarly, we omit the $\mathbf{0}$ and write $(\mathbf{y}_1, \ldots, \mathbf{y}_k) \xleftarrow{\$} \mathcal{D}^k_{\mathcal{R}_q, \sigma}$ to mean that a vector $\mathbf{y}$ is generated according to $\mathcal{D}_{\mathbb{Z}^{kn}, \mathbf{0}, \sigma}$ and then gets interpreted as k polynomials $\mathbf{y}_i$.

The definition of maximum singular values when working over the ring $\mathcal{R}_q$ is exactly the same as when working over $\mathbb{R}$. If $\mathbf{R} \in \mathcal{R}_q^{k \times m}$, then $\mathsf{s}_1(\mathbf{R}) = \max_{\mathbf{u} \in \mathcal{R}_q^m} \frac{\|\mathbf{Ru}\|}{\|\mathbf{u}\|}$. On rings too it holds that $\|\mathbf{Ru}\| \leq \mathsf{s}_1(\mathbf{R})\|\mathbf{u}\|$ for every $\mathbf{R} \in \mathcal{R}_q^{1 \times m}$ and $\mathbf{u} \in \mathcal{R}_q$. The following Lemma is a result by Ducas et al. [DM14].

Lemma 4 (Fact 6 in [DM14]). *If $\mathbf{M} \xleftarrow{\$} \mathcal{D}^{k \times m}_{R_q, s}$, then for the anticirculant representation of $\mathbf{M}$ with probability greater than $1 - 2\exp(-2n)$ it holds that $\mathsf{s}_1(\mathbf{M}) \leq \frac{s}{\sqrt{\pi}} \sqrt{n}(\sqrt{k} + \sqrt{m} + \log n)$.*

The following Theorem from [MP12] shows how a (pseudo-)random vector $\mathbf{U}$, for which no trapdoor is known, can be extended into a pseudo-random vector $[\mathbf{U}|\mathbf{V}]$ for which we will be able to sample from $\mathcal{D}^\perp_{[\mathbf{U}|\mathbf{V}+\mathbf{mG}], \mathbf{u}, \sigma}$ for any invertible $\mathbf{m}$ and for some standard deviation σ.

Theorem 1 (adapted from [MP12]). *Let $\mathbf{A}$ be a vector in $\mathcal{R}_q^{1 \times \ell}$ and $\mathbf{X}$ be a matrix in $\mathcal{R}_q^{\ell \times m}$. Also define the* gadget matrix $\mathbf{G} = [1 \; \lceil q^{1/m} \rceil \; \ldots \; \lceil q^{(m-1)/m} \rceil]$. *Then for any invertible $\mathbf{m} \in \mathcal{R}_q$, there is an algorithm that can sample from the distribution $\mathcal{D}^\perp_{[\mathbf{A}|\mathbf{AX}+\mathbf{mG}], \mathbf{u}, \sigma}$ for any $\sigma \sim q^{\frac{1}{m}} s_1(\mathbf{X}) > \tilde{\lambda}(\Lambda^\perp([\mathbf{A}|\mathbf{AX}+\mathbf{mG}]))$ for any $\mathbf{u} \in \mathcal{R}_q$.*

Lemma 5 is a combination of the double-trapdoor idea from [ABB10], with the sampling procedure in [BLP+13].

Lemma 5. *Suppose $\mathbf{U} \in \mathcal{R}_q^{1 \times k}$ and $\mathbf{V} \in \mathcal{R}_q^{1 \times m}$ are polynomial vectors, and $\mathbf{B}_U, \mathbf{B}_{(U,V)}$ are bases of $\Lambda^\perp(\mathbf{U})$ and $\Lambda^\perp([\mathbf{U}|\mathbf{V}])$ respectively such that $\|\tilde{\mathbf{B}}_U\|$,*

$\|\tilde{\mathbf{B}}_{(U,V)}\| < \sigma\sqrt{\pi/\ln(2n+4)}$.
Then, there exists an algorithm $\mathsf{SampleD}(\mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{u}, \sigma)$*, where* $\mathbf{B}$ *is either* $\mathbf{B}_U$ *or* $\mathbf{B}_{(U,V)}$*, that can efficiently sample from the distribution* $D^{\perp}_{[\mathbf{U}|\mathbf{V}],\mathbf{u},\sigma}$ *for any* $\mathbf{u} \in \mathcal{R}_q$.

2.5 Hard Problems

The security of our construction will be based on two well-studied lattice problems over rings: Ring-SIS and Ring-LWE.

Definition 1. (Ring-SIS$_\beta$ problem) *The Ring-SIS$_\beta$ problem is given a uniformly distributed vector* $\mathbf{A} \in \mathcal{R}_q^{1\times m}$ *to find a vector* $\mathbf{S} \in \mathcal{R}_q^{m+1}$ *such that* $[\mathbf{A}|\mathbf{1}]\mathbf{S} = \mathbf{0}$ *and* $\|\mathbf{S}\| \leq \beta$.

It was shown in Theorem 5.1 in [LM06] that there is a polynomial-time reduction from solving the shortest vector problem over the ring to Ring-SIS.

Definition 2. *The* Ring-LWE$_D$ distribution *outputs pairs* $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_q \times \mathcal{R}_q$ *such that* $\mathbf{b} = \mathbf{as} + \mathbf{e}$ *for a uniformly random* $\mathbf{a}$ *from* $\mathcal{R}_q$ *and* $\mathbf{s}, \mathbf{e}$ *sampled from distribution* D.

The Ring-LWE$_{k,D}$ decisional problem *on ring* $\mathcal{R}_q$ *with distribution* D *is to distinguish whether* k *pairs* $(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_k, \mathbf{b}_k)$ *were sampled from the Ring-LWE$_D$ distribution or from the uniform distribution over* $\mathcal{R}_q^2$.

In [LPR13] it was shown that there exists a polynomial-time quantum reduction from solving the shortest vector problem over the ring to Ring-LWE with Gaussian error distribution.

We use the root Hermite factor δ introduced in [GN08] to estimate the hardness for given parameters of the lattice problems in the security reductions. We will deduct the number of bits of security from it using the worst-case analysis by Akim et al. (cfr. Section 6 in [ADPS16]).

Finally, we recall the following lemma from [Lyu16]. It states that if the input set of a deterministic function is larger than the set of its output, there exists a collision with non-negligible probability.

Lemma 6 (Lemma 2.11 in [Lyu16]). *Let* $h : X \to Y$ *be a deterministic function where* X *and* Y *are finite sets and* $|X| \geq 2^{\lambda}|Y|$. *If* x *is chosen uniformly at random from* X*, with probability at least* $1 - 2^{-\lambda}$ *there exists another* $x' \in X$ *such that* $h(x) = h(x')$.

3 Relaxed Zero-Knowledge Proofs over Lattices

We define *relaxed* Σ*-protocols* and *relaxed non-interactive zero-knowledge proofs of knowledge* where the relaxed soundness definition guarantees the extraction of a witness from a wider language than the one used by an honest prover. Proofs with relaxed extracted notions have been used implicitly in previous work, e.g., for schemes based on discrete logarithms in group of unknown order [CKY09,CL03,CS03,XLL08] and some lattice-based schemes [Lyu12,LN17].

Camenisch et al. [CKY09] previously defined zero-knowledge proofs of knowledge with a very general relaxed extraction guarantee that in particular covers the peculiarities that arise when using the strong RSA assumption. We give a simpler definition here that suffices for the lattice-based protocols that we consider.

3.1 Definition of Relaxed Zero-Knowledge Proofs

Let $L \subseteq \{0,1\}^*$ be a language with witness relation R, meaning $x \in L \Leftrightarrow \exists\ w\ :\ (x,w) \in R$. Let $\bar{L} \supseteq L$ be a relaxed language with witness relation $\bar{R} \supseteq R$. We define *relaxed Σ-protocols* inspired by the definitions by Dåmgard [Dam02] and Faust et al. [FKMV12], but with a relaxed soundness condition that guarantees the extraction of a witness from $\bar{R}$ rather than R (similar to Camenisch et al. [CKY09]).

Definition 3 (Relaxed Σ-protocols). *A* relaxed Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ *for relations* $(R, \bar{R})$ *is a three-round public-coin interactive proof system where* $\mathcal{P} = (\mathcal{P}_0, \mathcal{P}_1)$ *and* $\mathcal{V} = (\mathcal{V}_0, \mathcal{V}_1)$ *are couples of PPT algorithms that, on top of the standard correctness and honest-verifier zero-knowledge (HVZK) properties recalled in Appendix A.1, satisfy the following property:*

Relaxed special soundness. *There exists an efficient algorithm* E*, called special extractor, that given two accepting conversations* (α, β, γ) *and* $(\alpha, \beta', \gamma')$ *for language member* $\bar{x} \in \bar{L}$ *where* $\beta \neq \beta'$*, computes* $\bar{w} \leftarrow \mathsf{E}(\bar{x}, \alpha, \beta, \gamma, \beta', \gamma')$ *such that* $(\bar{x}, \bar{w}) \in \bar{R}$*.*

Remark that relaxed Σ-protocols are relaxed proofs of knowledge, as the knowledge extractor extracts from $\mathcal{P}$ a pair (x, w) in $\bar{R}$ (the proof is a straightforward adaptation to relaxed protocols of the proof of Theorem 1 in [Dam02]).

Similarly to standard Σ-protocols, a relaxed Σ-protocol $(\mathcal{P}, \mathcal{V})$ can be turned into a relaxed non-interactive zero-knowledge (NIZK) proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$ using the Fiat-Shamir transform [FS87] that computes the second round $\beta \leftarrow \mathcal{H}_c(x, \alpha)$, where α is the first round of the proof and $\mathcal{H}_c$ is a random oracle.

Definition 4 (Relaxed NIZK). *A* relaxed NIZK proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$ *in the random-oracle model for relations* (R, R') *is couple of PPT algorithms that satisfy the standard correctness and unbounded zero-knowledge properties recalled in Appendix A.1, as well as the following soundness property:*

Relaxed unbounded simulation soundness. *There exists a PPT simulator* S *that simulates random-oracle responses as well as NIZK proofs, including for members* $x \notin L$*, such that for all PPT adversaries* A*,*

$$\Pr\left[\mathcal{V}^{\mathsf{S}_1}(x^*, \pi^*) = 1 \wedge x^* \notin \bar{L} \wedge (x^*, \pi^*) \notin Q\ :\ (x^*, \pi^*) \leftarrow \mathsf{A}^{\mathsf{S}}(1^\lambda)\right]$$

is negligible, where Q *is the set of tuples* (x, π) *where* A *made a query* $\mathsf{S}(x)$ *and obtained response* π*.*

Faust et al. [FKMV12] proved that the Fiat-Shamir transform of an HVZK Σ-protocol with *quasi-unique respones* yields an unbounded non-interactive zero-knowledge protocol in the random-oracle model. Since the Σ-protocols we consider do not always have quasi-unique responses, we suggest an alternative construction from one-time signature (OTS) schemes. We sketch the construction here; details and proofs can be found in Appendix B.

Given an interactive protocol $(\mathcal{P}, \mathcal{V})$ and a OTS scheme $(\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Vf})$, we construct a non-interactive zero-knowledge (NIZK) proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$ using a random oracle $\mathcal{H}_c$ with range equal to the space of the verifier's coins. The proving algorithm $\mathcal{P}^{\mathcal{H}_c}(x, w)$ computes a proof π by choosing random coins ρ, generating a OTS key pair $(sk, vk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$, and computing $\alpha \leftarrow \mathcal{P}_0(x, w; \rho)$. It determines the challenge as $\beta \leftarrow \mathcal{H}_c(x, \alpha, vk)$ and finally computes $\gamma \leftarrow \mathcal{P}_1(x, w, \alpha, \beta; \rho)$ and signs the whole transcript as $\sigma \leftarrow \mathsf{OTS.Sign}(sk, (x, \alpha, \beta, \gamma))$. The proof is $\pi = ((\alpha, vk), \beta, (\gamma, \sigma))$. Verification $\mathcal{V}^{\mathcal{H}_c}(x, \pi)$ checks that $\beta = \mathcal{H}_c(x, \alpha, vk)$, that $\mathcal{V}_1(x, \alpha, \beta, \gamma) = 1$, and that $\mathsf{OTS.Vf}(pk, \sigma, (x, \alpha, \beta, \gamma)) = 1$.

3.2 A Relaxed Σ-protocol to Prove Linear Relations

We rephrase Lyubashevsky's "Fiat-Shamir with aborts" technique [Lyu09,Lyu12] as a relaxed Σ-protocol for the languages $(L, \bar{L})$ associated to the following relations:

$$R = \left\{ ((\mathbf{A}, \mathbf{U}), (\mathbf{S}, \mathbf{1})) \in \mathcal{R}_q^{\ell \times m} \times \mathcal{R}_q^{1 \times \ell} \times \mathcal{R}_q^m \times \{\mathbf{1}\} : \begin{array}{c} \mathbf{AS} = \mathbf{U}, \|\mathbf{S}\| \leq N \\ \|\mathbf{S}\|_\infty < (q-1)/(2n) \end{array} \right\}$$

$$\bar{R} = \left\{ ((\mathbf{A}, \mathbf{U}), (\bar{\mathbf{S}}, \bar{\mathbf{c}})) \in \mathcal{R}_q^{\ell \times m} \times \mathcal{R}_q^{1 \times \ell} \times \mathcal{R}_q^m \times \bar{\mathcal{C}} : \begin{array}{c} \mathbf{A}\bar{\mathbf{S}} = \bar{\mathbf{c}}\mathbf{U}, \|\bar{\mathbf{S}}\| \leq \bar{N} \\ \|\bar{\mathbf{S}}\|_\infty \leq \bar{N}_\infty \end{array} \right\}$$

for some positive constants N, $\bar{N}$, $\bar{N}_\infty$ with $N \leq \bar{N}$. We set the challenge set to be $\mathcal{C} \subseteq \mathcal{R}_3^{(2^{K_c})}$ and the set of relaxed challenges to be $\bar{\mathcal{C}} \subseteq \mathcal{R}_5^{(2^{K_c})}$, $K_c > 0$. Let C (resp. $\bar{C}$) be a bound on $\|\mathbf{c}\|$ for $\mathbf{c} \in \mathcal{C}$ (resp. $\bar{\mathbf{c}} \in \bar{\mathcal{C}}$). Finding a witness $(\bar{\mathbf{S}}, \bar{\mathbf{c}})$ for an element $(\mathbf{A}, \mathbf{U})$ of the language $\bar{L}$ is hard under the computational assumption that Ring-SIS$_\beta$ is hard, where $\beta = \sqrt{(\bar{N}^2 + \bar{C}^2)}$.

The Σ-protocol $(\mathcal{P}, \mathcal{V})$ works as follows. First, the prover $\mathcal{P}$ samples a masking vector $\mathbf{Y} \xleftarrow{\$} \mathcal{D}_\sigma^m$ (we will determine the value of σ in a moment), and sends $\mathbf{T} = \mathbf{AY}$ to $\mathcal{V}$. Next, the verifier $\mathcal{V}$ samples a challenge $\mathbf{c} \in \mathcal{C}$ and sends it back to $\mathcal{P}$. The prover then constructs $\mathbf{Z} = \mathbf{Y} + \mathbf{cS}$ and, depending on rejection sampling (see Theorem 4.6 in [Lyu12]), either aborts or sends it to $\mathcal{V}$. The verifier accepts if $\mathbf{AZ} - \mathbf{cU} = \mathbf{T}$ and $\|\mathbf{Z}\| \leq 1.05\sigma\sqrt{nm} =: N_2$, $\|\mathbf{Z}\|_\infty \leq 8\sigma =: N_\infty$. Now, observe that the zero-knowledge property is guaranteed by rejection sampling. A standard deviation $\sigma = aT$, where $T = C \cdot N\sqrt{n}$ is a bound on the norm of $\mathbf{cS}$ obtained from Lemma 2 and $a > 0$, guarantees that the prover outputs something with probability greater than $(1 - 2^{100})/e^{12/a + 1/(2a^2)}$ (cfr. Theorem 4.6 in [Lyu12]). Finally, we set $\bar{N} = 2N = 2.1\sigma\sqrt{nm}$ and $\bar{N}_\infty = 2N_\infty = 16\sigma$. In the following theorem we prove that our protocol is a relaxed Σ-protocol.

Theorem 2. *The protocol described above is a relaxed Σ-protocol for relations $(R, \bar{R})$.*

Proof. Correctness follows from Lemma 3. Zero-knowledge follows from rejection sampling: a simulator S can simply sample $\mathbf{Z} \xleftarrow{\$} \mathcal{D}^m_{\mathcal{R}_q,\sigma}$, $\mathbf{c} \xleftarrow{\$} \mathcal{C}$ and set $\mathbf{T} := \mathbf{AZ} - \mathbf{cU}$. Finally, special soundness is proved as usual by defining an extractor that runs $\mathcal{P}$ twice on different challenges and output as response the difference of the responses. □

3.3 Proving Knowledge of Bounded-Degree Secrets in a Subring

In our construction of an anonymous attribute token scheme, we will use the above protocol in a modified form to let a prover prove knowledge of a $[\mathbf{m}; \mathbf{s}]$ where $\mathbf{m}$ is a small element in a subring $\mathcal{R}_q^{(2^{K_m})}$ of $\mathcal{R}_q$ and with degree $\deg(\mathbf{m}) < d$ for some constant $d < n$. The fact that $\mathbf{m}$ is in the subring can be proved by exploiting the subring structure. Indeed, the challenge space $\mathcal{C} = \mathcal{R}_3^{(2^{K_c})}$ is a subset of $\mathcal{R}_q^{(2^{K_m})}$ when $K_m \geq K_c$. To have the largest possible set of challenges, we set $K_c = K_m$. By also sampling the first component $\mathbf{y}_m$ of the "masking" vector $\mathbf{Y} = [\mathbf{y}_m; \mathbf{y}_s]$ from the subring $\mathcal{R}_q^{(2^{K_m})}$, the output vector $[\mathbf{z}_m; \mathbf{z}_s] = [\mathbf{y}_m; \mathbf{y}_s] + \mathbf{c}[\mathbf{m}; \mathbf{s}]$ will be such that $\mathbf{z}_m \in \mathcal{R}_q^{(2^{K_m})}$. Sampling a discrete Gaussian distribution from the subring $\mathcal{R}_q^{(2^{K_m})}$ can be done by sampling from $\mathcal{D}_{\mathbb{Z}^{2^{K_m}},\sigma}$ and mapping the 2^{K_m} coordinates into the non-zero coefficients of the polynomials. The zero-knowledge property remains guaranteed by rejection sampling.

Proving that $\mathbf{m}$ is of degree strictly less than $d < n$ can be done by carefully choosing the challenge set and the domain of the masking vector. In particular, if $\deg(\mathbf{m}) \leq d_m$ and challenges are chosen to be polynomials of degree d_c such that $d_c + d_m < d$, then $\deg(\mathbf{mc}) < d$. Letting the prover sample the masking vector $\mathbf{y}_m$ from the polynomials of degree less than d and applying rejection sampling as usual preserves the zero-knowledge property when computing $\mathbf{z}_m = \mathbf{mc} + \mathbf{y}_m$. By letting the verifier additionally check that $\deg(\mathbf{z}_m) < d$, the extractor is guaranteed to be able to extract a witness $\bar{\mathbf{m}} = \mathbf{z}_{m,1} - \mathbf{z}_{m,2}$ of degree strictly less than d.

Note that sampling a discrete Gaussian distributions of polynomials of degree at most $d-1$ from the subring $\mathcal{R}_q^{(2^{K_m})}$ can be done by sampling from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$ for $m = \lfloor (d-1)n/2^{K_m} \rfloor$ and mapping coordinates to coefficients. To have a clearer notation, we define $\mathcal{Y}_d$ to be the set of elements in the subring $\mathcal{R}_q^{(2^{K_m})}$ with degree at most $d-1$, so that the distribution of the full masking vector $\mathbf{Y}$ can be written as $\mathcal{D}_{\mathcal{Y} \times \mathcal{R}_q,\sigma}$.

A possible drawback of this technique, however, is that it shrinks the size of the challenge space, so that the proof may have to be repeated several times to obtain soundness.

4 A Relaxed Lattice-Based Commitment Scheme

We describe a commitment scheme with an efficient proof of knowledge of a committed message using the Σ-protocol of Section 3. To compensate for relaxed extraction properties of the Σ-protocol, we define *relaxed commitments*, where the opening algorithm accepts messages and opening information that would not be accepted as input, respectively produced as output, by the honest commitment algorithm. We define a correspondingly relaxed binding property that divides messages into classes and only considers binding attacks for messages belonging to different classes.

4.1 Definition of Relaxed Commitments

A relaxed commitment scheme C for message space $\mathcal{U}$ and relaxed message space $\bar{\mathcal{U}} \supseteq \mathcal{U}$ consists of a triple of algorithms $(\mathsf{ComParGen}, \mathsf{Commit}, \mathsf{OpenVf})$, where $cpar \leftarrow \mathsf{ComParGen}(\mathcal{U}, 1^\lambda)$ generates the parameters on input the message space and the security parameter, $(c, o) \leftarrow \mathsf{Commit}(cpar, M)$ computes the commitment value c and the opening information o on input the parameters and a message in $\mathcal{U}$, and $\{1, 0\} \leftarrow \mathsf{OpenVf}(cpar, c, \bar{M}, \bar{o})$ verifies whether $\bar{o}$ is an opening of $\bar{M} \in \bar{\mathcal{U}}$ for the commitment c.

A commitment scheme must satisfy the standard correctness and hiding properties that are recalled in Appendix A.2. The binding property is relaxed by considering a partition of the relaxed message space $\bar{\mathcal{U}}$ and considering only attacks where the adversary can open a commitment to two messages coming from different components of the partition. The partition is defined by the message relaxation function $f : \mathcal{U} \to 2^{\bar{\mathcal{U}}}$ that maps a message $M \in \mathcal{U}$ to a partition component $f(M) \subseteq \bar{\mathcal{U}}$. We say that the commitment scheme is *f-relaxed binding* if no adversary can open a commitment to two messages from different components.

Definition 5 (Relaxed Binding). *A relaxed commitment scheme C is f-binding for a function $f : \mathcal{U} \mapsto 2^{\bar{\mathcal{U}}}$ if for all polynomial-time A*

$$\mathsf{Pr}\left[\begin{array}{c} \mathsf{OpenVf}(cpar, c, \bar{M}_0, \bar{o}_0) = 1 \\ \wedge\, \mathsf{OpenVf}(cpar, c, \bar{M}_1, \bar{o}_1) = 1 \\ \wedge\, \nexists\, M \in \mathcal{U} \,:\, \{\bar{M}_0, \bar{M}_1\} \subseteq f(M) \end{array} : \begin{array}{c} cpar \leftarrow \mathsf{ComParGen}(\mathcal{U}, 1^\lambda), \\ (c, \bar{M}_0, \bar{o}_0, \bar{M}_1, \bar{o}_1) \leftarrow \mathsf{A}(cpar) \end{array}\right] \leq \nu(n)\,.$$

4.2 Message and Challenge Spaces

Our goal is to create a commitment scheme where the relaxed Σ-protocol from Section 3 can be used to prove knowledge of a committed message, where the message and opening information are part of the witness. The problem with relaxed Σ-protocols is that they cannot guarantee the extraction of a valid witness for the original relation R, but only for the relaxed relation $\bar{R}$. The witnesses in the latter have larger norms and explicitly admit "small multiples": if $(\mathbf{S}, \mathbf{1})$ is a valid witness in R so that $\mathbf{AS} = \mathbf{U}$, then $(\bar{\mathbf{S}} = \bar{\mathbf{c}}\mathbf{S}, \bar{\mathbf{c}})$ is a valid witness in $\bar{R}$ so that $\mathbf{A}\bar{\mathbf{S}} = \bar{\mathbf{c}}\mathbf{U}$, where $\bar{\mathbf{c}} \in \bar{\mathcal{C}} = \{\mathbf{c} - \mathbf{c}' \,:\, \mathbf{c}, \mathbf{c}' \in \mathcal{C}\}$ and where $\mathcal{C}$ is the challenge space.

By relaxing the opening verification of the commitment scheme to accept extracted messages and opening information, we allow a commitment to be opened to a small multiple $\bar{\mathbf{c}}\mathbf{m}$ of the originally committed message $\mathbf{m} \in \mathcal{U}$. In order to preserve a meaningful notion of relaxed binding, we must choose the message and challenge spaces so that the sets of small multiples of different messages are disjoint, i.e., that there do not exist distinct $\mathbf{m}, \mathbf{m}' \in \mathcal{U}$ and $\mathbf{c}, \mathbf{c}' \in \bar{\mathcal{C}}$ such that $\mathbf{mc} = \mathbf{m}'\mathbf{c}'$.

For efficiency reasons, we choose messages and challenges from the subring $\mathcal{R}_3^{(2^{K_m})}$ so that they have at most 2^{K_m} nonzero coefficients. By choosing the message and challenge spaces as

$$\begin{aligned}
\mathcal{U} &= \{\mathbf{1}\} \cup \{\mathbf{m} \in \mathcal{R}_3^{(2^{K_m})} \ : \ \deg(\mathbf{m}) = n/2 \ \wedge \ \mathbf{m} \text{ is irreducible in } \mathbb{Z}_q[\mathbf{x}]\} \\
\mathcal{C} &= \{\mathbf{c} \in \mathcal{R}_3^{(2^{K_m})} \ : \ \deg(\mathbf{c}) < n/4\} \\
\bar{\mathcal{U}} &= \{\bar{\mathbf{m}} \in \mathcal{R}_{2p+1}^{(2^{K_m})} \ : \ \deg(\bar{\mathbf{m}}) < 3n/4\} \\
\bar{\mathcal{C}} &= \{\mathbf{c} - \mathbf{c}' \ : \ \mathbf{c}, \mathbf{c}' \in \mathcal{C}\} \ ,
\end{aligned} \tag{1}$$

we have that each $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$ can have at most one irreducible factor of degree $n/2$ in $\mathbb{Z}_q[\mathbf{x}]$. By defining the message relaxation function f as

$$\begin{aligned}
f(\mathbf{m}) &= \{\bar{\mathbf{m}} \in \bar{\mathcal{U}} \ : \ \mathbf{m} | \bar{\mathbf{m}} \text{ in } \mathbb{Z}_q[\mathbf{x}]\} \quad \text{for } \mathbf{m} \neq \mathbf{1} \\
f(\mathbf{1}) &= \{\bar{\mathbf{m}} \in \bar{\mathcal{U}} \ : \ \nexists \, \mathbf{m} \in \mathcal{U} \setminus \{\mathbf{1}\} \ : \ \mathbf{m} | \bar{\mathbf{m}} \text{ in } \mathbb{Z}_q[\mathbf{x}]\} \ ,
\end{aligned} \tag{2}$$

the unique factorization of polynomials in $\mathbb{Z}_q[\mathbf{x}]$ guarantees that the partition components $f(\mathbf{m})$ and $f(\mathbf{m}')$ are disjoint for any distinct $\mathbf{m}, \mathbf{m}' \in \mathcal{U}$.

To generate elements of $\mathcal{U}$, we suggest to generate random monic polynomials of degree $n/2$ in $\mathcal{R}_3^{(2^{K_m})}$ and test them for irreducibility, which can be done efficiently (e.g., using Proposition 3.4.4 in [Coh13]). By the Gauss' formula, the number of monic polynomials of degree $n/2$ that are irreducible in $\mathbb{Z}_q[\mathbf{x}]$ is approximately $q^{n/2}/(n/2)$. Assuming that the irreducible polynomials are "spread evenly" across $\mathbb{Z}_q[\mathbf{x}]$, one expects to sample an average of $n/2$ polynomials until finding an irreducible one.

4.3 Lattice-based Relaxed Commitment Scheme

Our relaxed commitment uses message space $\mathcal{U}$ and relaxed message space $\bar{\mathcal{U}}$ defined in Equation (1). The algorithms of our relaxed commitment scheme rC are as follows.

Parameter generation. $\mathsf{ComParGen}$ selects a uniformly random commitment key $\mathbf{C} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ and the parameters $\bar{N}_c$ and $\bar{N}_{c,\infty}$ that will be defined in Section 6. It outputs $cpar = (\mathbf{C}, \bar{N}_c, \bar{N}_{c,\infty})$.

Commitment generation. On input $(cpar, \mathbf{m})$, the algorithm Commit first checks that $\mathbf{m} \in \mathcal{R}_3^{(2^{K_m})}$, that $\deg(\mathbf{m}) = n/2$, and that $\mathbf{m}$ is irreducible. It then selects uniformly random $\mathbf{E} \xleftarrow{\$} \mathcal{R}_3^{1\times m}$ and $\mathbf{b} \xleftarrow{\$} \mathcal{R}_3$, and constructs the commitment as $\mathbf{F} = (\mathbf{C} + \mathbf{mG} + \mathbf{E})\mathbf{b}^{-1}$. It outputs $(\mathbf{F}, (\mathbf{1}, \mathbf{E}, \mathbf{b}))$.

Opening verification. On input a message $\bar{\mathbf{m}}$, a commitment $\mathbf{F}$, and opening values $(\bar{\mathbf{c}}, \bar{\mathbf{E}}, \bar{\mathbf{b}})$, OpenVf outputs 1 if $\mathbf{F} = (\bar{\mathbf{c}}\mathbf{C} + \bar{\mathbf{m}}\mathbf{G} + \bar{\mathbf{E}})\bar{\mathbf{b}}^{-1}$, $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$ and

$(\bar{\mathbf{c}}, \bar{\mathbf{E}}, \bar{\mathbf{b}}) \in \mathcal{OV} = \{(\bar{\mathbf{c}}, \bar{\mathbf{E}}, \bar{\mathbf{b}}) \in \bar{\mathcal{C}} \times \mathcal{R}_q^{1\times m} \times \mathcal{R}_q : \|[\bar{\mathbf{E}}, \bar{\mathbf{b}}]\| \leq \bar{N}_c \wedge \|[\bar{\mathbf{E}}, \bar{\mathbf{b}}]\|_\infty \leq \bar{N}_{c,\infty}\}$.

It is easy to see that our construction satisfies correctness. We prove the hiding property under a new assumption, defined as Assumption 1 below. To gain trust in this assumption, we also give a selective variant in Assumption 2 that we show to be equivalent to Ring-LWE and that, through a complexity leveraging argument, implies Assumption 1.

Assumption 1 *Consider the following game between an adversary* A *and a challenger for fixed* $m \in \mathbb{N}$ *and distribution* D*:*

1. *The challenger outputs a uniformly random* $\mathbf{C} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *to* A*.*
2. A *sends back* $\mathbf{m} \in \mathcal{U}$*.*
3. *The challenger samples a uniformly random bit* $b \xleftarrow{\$} \{0,1\}$*. If* $b = 1$*, it samples an error vector* $\mathbf{E} \xleftarrow{\$} D^m$ *and a uniform secret* $\mathbf{s} \xleftarrow{\$} D$*, and sends* $\mathbf{F} = (\mathbf{C} + \mathbf{mG} - \mathbf{E})\mathbf{s}^{-1}$ *to* A*. Otherwise, it sends a uniform* $\mathbf{F} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *to* A*.*
4. A *sends a bit* b' *to the challenger.*

The advantage of A *in winning the game is* $\left|\mathsf{Pr}(b = b') - \frac{1}{2}\right|$*. The assumption states that no PPT* A *can win the previous game with non-negligible advantage.*

Assumption 2 (Selective variant of Assumption 1.) *Consider the game of Assumption 1, but with steps 1 and 2 switched, meaning,* A *outputs* $\mathbf{m} \in \mathcal{U}$ *before being given* $\mathbf{C}$*. The assumption states that no PPT adversary can win this previous game with non-negligible advantage.*

Theorem 3. *Assumption 2 holds for* $m \in \mathbb{N}$ *and distribution* D *if the Ring-LWE*$_{m,D}$ *assumption holds.*

Proof. Let A be an attacker breaking Assumption 2. Then the following algorithm B breaks Ring-LWE$_{m,D}$ in essentially the same time and with the same advantage as A. Upon input a challenge Ring-LWE instance $(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_m, \mathbf{b}_m)$, algorithm B runs A to obtain $\mathbf{m}$. It then sets $\mathbf{A} = [\mathbf{a}_1; \ldots; \mathbf{a}_m]$, $\mathbf{B} = [\mathbf{b}_1; \ldots; \mathbf{b}_m]$, $\mathbf{C} = \mathbf{B} - \mathbf{mG}$ and $\mathbf{F} = \mathbf{A}$, and feeds $(\mathbf{B}, \mathbf{F})$ back to A. When A outputs $b' = 1$, then B decides that its input came from Ring-LWE$_{m,D}$, otherwise that it was uniform.

Note that if $(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_m, \mathbf{b}_m)$ come from the uniform distribution over $\mathcal{R}_q^2$, then also $\mathbf{C}$ and $\mathbf{F}$ are uniformly distributed in $\mathcal{R}_q^{1\times m}$. If they come from the Ring-LWE distribution, however, then there exist an $\mathbf{s}$ and $\mathbf{E}$ sampled from D and D^m, respectively, such that $\mathbf{As} + \mathbf{E} = \mathbf{B}$. Therefore, $\mathbf{A} = \frac{\mathbf{B}-\mathbf{E}}{\mathbf{s}} = \frac{\mathbf{C}+\mathbf{mG}-\mathbf{E}}{\mathbf{s}}$. □

The proof of the following theorem follows from a straightforward complexity leveraging argument by guessing the value of $\mathbf{m} \in \mathcal{U}$.

Theorem 4. *Let* A *be a PPT algorithm that has advantage* ϵ *in breaking Assumption 1 in time* t*. Then there exists a PPT algorithm* B *with running time* t *and advantage* $\frac{\epsilon}{|\mathcal{U}|}$ *in breaking Assumption 2.*

We are now ready to prove the hiding property of the commitment scheme under Assumption 1.

Theorem 5 (Hiding). *The relaxed commitment scheme above is computationally hiding when Assumption 1 holds for m and the uniform distribution over $\mathcal{R}_3$.*

Proof. Given A breaking the hiding property of the commitment scheme, consider the following adversary B in the game of Assumption 1. Upon receiving $\mathbf{C} \in \mathcal{R}_q^{1\times m}$ from the challenger, B sends it to A as part of the public parameters *cpar*. When A sends challenge messages $\mathbf{m}_0$, $\mathbf{m}_1$, B samples a random bit $b \xleftarrow{\$} \{0,1\}$ and sends $\mathbf{m}_b$ to the challenger. Upon receiving $\mathbf{F}$, B sends it to A as the commitment. When A outputs a bit $b' = b$, B outputs $b'' = 1$ to the challenger, otherwise $b'' = 0$. It is clear that when B's input $\mathbf{F}$ is uniform, then A's view is independent of b, so that A has zero advantage guessing b, while if B's input is based on $\mathbf{m}_b$, it is distributed exactly as a commitment of $\mathbf{m}_b$. The advantage of B in breaking Assumption 1 is therefore half the advantage of A in breaking the hiding property. □

The following theorem shows that the relaxed binding property holds under the Ring-SIS assumption.

Theorem 6 (Relaxed Binding). *The relaxed commitment scheme above is f-relaxed binding for the function f in Equation (2) if the Ring-SIS$_{\beta_c}$ assumption holds.*

Proof. Assume that there exists a PPT algorithm A that breaks the f-relaxed binding property. Consider the algorithm B that solves the Ring-SIS$_{\beta_c}$ problem as follows. On input $[\mathbf{A}|\mathbf{1}] \in \mathcal{R}_q^{1\times(m+1)}$, algorithm B runs A on input parameters *cpar* that include $\mathbf{C} = \mathbf{A}$. When A outputs a commitment $\mathbf{F}$, two distinct messages $\bar{\mathbf{m}}_0, \bar{\mathbf{m}}_1 \in \bar{\mathcal{U}}$, and two openings $(\bar{\mathbf{c}}_0, \bar{\mathbf{E}}_0, \bar{\mathbf{b}}_0)$ and $(\bar{\mathbf{c}}_1, \bar{\mathbf{E}}_1, \bar{\mathbf{b}}_1)$ such that $\mathsf{OpenVf}(\mathit{cpar}, \mathbf{F}, \bar{\mathbf{m}}_i, (\bar{\mathbf{c}}_i, \bar{\mathbf{E}}_i, \bar{\mathbf{b}}_i)) = 1$ for $i = 0, 1$. We have that

$$\mathbf{F} \;=\; (\bar{\mathbf{c}}_0\mathbf{C} + \bar{\mathbf{m}}_0\mathbf{G} + \bar{\mathbf{E}}_0)\bar{\mathbf{b}}_0^{-1} \;=\; (\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}_1\mathbf{G} + \bar{\mathbf{E}}_1)\bar{\mathbf{b}}_1^{-1}$$

or, by rearranging terms, that

$$(\bar{\mathbf{b}}_1\bar{\mathbf{c}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{c}}_1)\mathbf{A} + (\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1)\mathbf{G} + \bar{\mathbf{b}}_1\bar{\mathbf{E}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{E}}_1 \;=\; \mathbf{0}\ . \tag{3}$$

Recalling that $\mathbf{G} = [1\ \lceil q^{1/m}\rceil\ \ldots\ \lceil q^{(m-1)/m}\rceil]$, the first component of the above vector is

$$(\bar{\mathbf{b}}_1\bar{\mathbf{c}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{c}}_1)\mathbf{a}_1 + \bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 + \bar{\mathbf{b}}_1\bar{\mathbf{e}}_{0,1} - \bar{\mathbf{b}}_0\bar{\mathbf{e}}_{1,1} \;=\; 0\ .$$

where $\mathbf{a}_1$, $\bar{\mathbf{e}}_{0,1}$, and $\bar{\mathbf{e}}_{1,1}$ are the first components of $\mathbf{A}$, $\bar{\mathbf{E}}_0$, and $\bar{\mathbf{E}}_1$, respectively. By setting $\mathbf{S} = [\bar{\mathbf{b}}_1\bar{\mathbf{c}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{c}}_1\ ;\ 0\ ;\ \ldots\ ;\ 0\ ;\ \bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 + \bar{\mathbf{b}}_1\bar{\mathbf{e}}_{0,1} - \bar{\mathbf{b}}_0\bar{\mathbf{e}}_{1,1}]$, we have that $[\mathbf{A}|\mathbf{1}]\mathbf{S} = 0$, as required, so B outputs $\mathbf{S}$ as its Ring-SIS$_{\beta_c}$ solution.

We have left to show that $\mathbf{S} \neq \mathbf{0}_m$ and that $\|\mathbf{S}\| \leq \beta_c$. First, assume that $\mathbf{S} = \mathbf{0}_m$. This would mean in particular that $\bar{\mathbf{b}}_1\bar{\mathbf{c}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{c}}_1 = \mathbf{0}$. Therefore, from Equation (3) it follows that

$$(\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1)\mathbf{G} + \bar{\mathbf{b}}_1\bar{\mathbf{E}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{E}}_1 \;=\; \mathbf{0}_m\ . \tag{4}$$

In this equation, we show that $\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 \neq \mathbf{0}$. Indeed, if $\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 = \mathbf{0}$, then multiplying both sides with $\bar{\mathbf{c}}_0$ and substituting $\bar{\mathbf{b}}_1\bar{\mathbf{c}}_0 = \bar{\mathbf{b}}_0\bar{\mathbf{c}}_1$ yields $\bar{\mathbf{b}}_0(\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1) = \mathbf{0}$, implying that $\bar{\mathbf{b}}_0 = \mathbf{0}$ or $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0 = \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$. The former is not possible because $\bar{\mathbf{b}}_0^{-1}$ must exist in order to pass the opening verification algorithm. The latter is impossible as well, because $\bar{\mathbf{c}}_0, \bar{\mathbf{c}}_1$ are polynomials of degree less than $n/4$, while $\bar{\mathbf{m}}_0, \bar{\mathbf{m}}_1 \in \bar{\mathcal{U}}$ are of degree less than $3n/4$, so that their products $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0, \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$ are of degree less than n. Therefore, if $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0 = \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$ in $\mathcal{R}_q = \mathbb{Z}_q[\mathbf{x}]/\langle \mathbf{x}^n + 1\rangle$, then also $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0 = \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$ in $\mathbb{Z}_q[\mathbf{x}]$. To be a valid f-binding attack, there cannot exist an $\mathbf{m} \in \mathcal{U}$ so that $\{\bar{\mathbf{m}}_0, \bar{\mathbf{m}}_1\} \in f(\mathbf{m})$. This implies that at least one message $\bar{\mathbf{m}}_b \in \{\bar{\mathbf{m}}_0, \bar{\mathbf{m}}_1\}$ has an irreducible divisor $\mathbf{m}$ of degree $n/2$ that doesn't divide $\bar{\mathbf{m}}_{1-b}$. Since $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0$ and $\bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$ are polynomials of degree less than n and $\mathbb{Z}_q[x]$ is a unique factorization domain, it must hold that $\bar{\mathbf{c}}_1\bar{\mathbf{m}}_0 \neq \bar{\mathbf{c}}_0\bar{\mathbf{m}}_1$, and thereby that $\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 \neq 0$ in Equation (4).

Letting $\mathbf{g}_i = \lceil q^{(i-1)/m} \rceil$, $i = 1, \ldots, m$, we can rearrange Equation (4) and consider its components

$$(\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1)\mathbf{g}_i \;=\; \bar{\mathbf{b}}_0\bar{\mathbf{e}}_{1,i} - \bar{\mathbf{b}}_1\bar{\mathbf{e}}_{0,i}\;. \tag{5}$$

Applying Lemma 2 for all $i = 1, \ldots, m$ it holds $\|\bar{\mathbf{b}}_0\bar{\mathbf{e}}_{1,i} - \bar{\mathbf{b}}_1\bar{\mathbf{e}}_{0,i}\| \leq 2\bar{N}_c^2\sqrt{n}$, as we know from the definition of $\mathcal{OV}$ that the infinity norm of $\mathbf{E}$ and $\mathbf{b}$ should be less than $\bar{N}_c$. Since $\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1 \neq 0$, it must have at least one non-zero coefficient $a_j\mathbf{x}^j$. Let i be such that $q^{(m-i)/m} \leq a_j \leq q^{(m-i+1)/m}$. Then the coefficient of $\mathbf{x}^j$ in the left-hand side of Equation (5) for this component i is $a_j\mathbf{g}_i \geq q^{(m-i)/m}q^{(i-1)/m} = q^{(m-1)/m}$, so that $\|(\bar{\mathbf{b}}_1\bar{\mathbf{m}}_0 - \bar{\mathbf{b}}_0\bar{\mathbf{m}}_1)\mathbf{G}\|_\infty \geq q^{(m-1)/m}$. Setting $\bar{N}_c$ and m using the parameters in Table 7.3 we reach a contradiction, showing that $\mathbf{S} \neq \mathbf{0}_m$.

Finally, we have to bound the norm of $\mathbf{S}$. Applying Lemma 2 and recalling that to be a valid opening value $\|[\mathbf{E}, \mathbf{b}]\| \leq \bar{N}_c$ we obtain that:

$$\|\mathbf{S}\| \leq \sqrt{2(\bar{N}_c^2\sqrt{n})^2 + 2(\bar{N}_{c,\infty}\sqrt{n} \cdot p\sqrt{2^{K_m} \cdot 3/4}\sqrt{n})^2} =: \beta_c$$

□

5 Relaxed Lattice-Based Signatures

We now introduce a signature scheme for which the $\mathsf{r}\Sigma$ protocol from Section 3 can be used to prove knowledge of a signature on a committed message. Similarly to the relaxed commitments of the previous section, we also define *relaxed signature schemes* to accommodate for the relaxed extraction of the $\mathsf{r}\Sigma$ protocol. More specifically, the verification algorithm is relaxed to accept messages and signatures that could never be signed, respectively produced, by the honest signing algorithm. At the same time, we also relax the unforgeability notion so that the adversary's forgery cannot be on a message that is within the span, through a function f, of its previous signing queries.

5.1 Definition of Relaxed Signatures

A relaxed signature scheme associated with message space $\mathcal{M}$ and relaxed messages space $\bar{\mathcal{M}} \supseteq \mathcal{M}$ consists of a parameter generation algorithm $\mathsf{SignParGen}$ that on input security parameter 1^λ outputs system parameters $spar$; a key generation algorithm $\mathsf{SignKeyGen}$ that on input $spar$ outputs a signing key sk and a verification key vk; a signing algorithm Sign that on input sk and a message $M \in \mathcal{M}$ outputs a signature sig; and a verification algorithm SignVf that on input vk, a message $\bar{M} \in \bar{\mathcal{M}}$ and a signature $\bar{sig}$ returns 1 if the signature is valid or 0 if it is invalid. Correctness requires that $\mathsf{SignVf}(vk, M, sig) = 1$ for all messages $M \in \mathcal{M}$, for all security parameters $\lambda \in \mathbb{N}$, for all $(sk, vk) \in \mathsf{SignKeyGen}(spar)$, and for all $sig \in \mathsf{Sign}(sk, M)$.

Relaxed unforgeability is parameterized by a message relaxation function $g : \mathcal{M} \to 2^{\bar{\mathcal{M}}}$.The adversary in the *g-relaxed unforgeability* below wins the game if it can output a valid signature on a message $\bar{M} \in \bar{\mathcal{M}}$ that is not in the span through g of its signature queries.

Definition 6 (Relaxed Unforgeability). *A relaxed signature scheme* $(\mathsf{SignParGen}, \mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{SignVf})$ *is g-relaxed unforgeable if for all PPT* A *the probability*

$$\mathsf{Pr}\left[\begin{array}{c|c} \mathsf{SignVf}\left(vk, \bar{M}, \bar{sig}\right) = 1 \\ \wedge\ \bar{M} \notin g(\mathcal{Q}) \end{array} : \begin{array}{c} spar \leftarrow \mathsf{SignParGen}(1^n), \\ (sk, vk) \leftarrow \mathsf{SignKeyGen}(spar), \\ (\bar{M}, \bar{sig}) \leftarrow \mathsf{A}^{\mathcal{O}_S}(n, spar, vk) \end{array}\right]$$

is negligible, where the oracle $\mathcal{O}_S(M)$ *returns* $\mathsf{Sign}(spar, sk, vk, M)$ *and* $\mathcal{Q}$ *is the set of* A*'s queries to* $\mathcal{O}_S$*.*

The concept of relaxed signatures is somewhat reminiscent of a technique used for proofs of knowledge of a strong-RSA-based signature in groups of unknown order [CL03]. Here, one has to prove that the message lies in a certain space, but the correctness of such a proof is only guaranteed when the actual message lies in a smaller interval. The approach was used in several privacy-preserving protocols, but was never formalized and did not require an adapted unforgeability notion.

5.2 Lattice-Based Relaxed Signature Scheme

We describe a relaxed signature scheme with message space $\mathcal{M} = \{(\mathbf{m}, \alpha) \in \mathcal{U} \times \{0,1\}^*\}$, where $\mathcal{U}$ is as defined in Equation (1). In a typical use case, $\mathbf{m}$ is a user identity and α an attribute value assigned to that user. Our scheme combines a weakly secure version of Boyen signatures [Boy10] to sign user identities and Gentry-Peikert-Vaikuntanathan signatures [GPV08] to sign attribute values.

To use the $\mathsf{r\Sigma}$ protocol from Section 3.2 to prove knowledge of a signature for a committed user identity $\mathbf{m}$, we relax the verification algorithm so that the (relaxed) witness that can be extracted from a valid $\mathsf{r\Sigma}$ protocol is still considered a valid signature for a message from the relaxed message space $\bar{M} = \bar{\mathcal{U}} \times \{0,1\}^*$, where $\bar{\mathcal{U}}$ is as defined in Equation (1).

Our relaxed signature scheme rS is described as follows:

System parameters. The system parameters *spar* include a uniformly random matrix $\mathbf{C} \in \mathcal{R}_q^{1\times m}$, a gadget vector $\mathbf{G}$ of length m as defined in Theorem 1, and a hash function $\mathcal{H} : \{0,1\}^* \to \mathcal{R}_q$. It also contains the parameters listed below; concrete values for these parameters will be provided in the correctness discussion and in Table 7.3.

- σ_t is the standard deviation of the trapdoor distribution,
- σ is the standard deviation of the signature distribution,
- p is a bound on the norm of user identities $\bar{\mathbf{m}}$,
- N_s is a bound on the norm of honestly created signatures,
- $\bar{N}_s$, $\bar{N}_{s,\infty}$, and $\bar{C}$ are bounds on the norm of components of signatures accepted by the relaxed verification algorithm,
- $\mathcal{C}$, and $\bar{\mathcal{C}}$ are challenge spaces defined in Equation (1).

When discussing the correctness of the signature, we give precise formulas for all the previous parameters but $\bar{N}_s$, $\bar{N}_{s,\infty}$, and $\bar{C}$. These last three will be discussed in Section 6. For correctness to hold, we only need to impose that $\bar{N}_s > N_s$ and $\bar{C} \geq 1$.

Key generation. The signer chooses a uniform polynomial $\mathbf{a} \in \mathcal{R}_q$ and sets $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$. The secret signing key is sampled as $\mathbf{X} \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q,\sigma_t}$. Letting $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$, the public verification key is the vector $\mathbf{V} = [\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}] = [\mathbf{A}|\mathbf{A}\mathbf{X} + \mathbf{G}|\mathbf{C}|\mathbf{1}] \in \mathcal{R}_q^{1\times(3+2m)}$.

Signing. If $M = (\mathbf{m}, \alpha) \notin \mathcal{M}$ then abort. Otherwise, the signer calculates $\mathbf{S} \leftarrow \mathsf{SampleD}([\mathbf{A}|\mathbf{B}|\mathbf{C} + \mathbf{m}\mathbf{G}], \mathcal{H}(\alpha), \sigma)$ (see Lemma 5) and outputs a signature $sig = (\mathbf{1}, [\mathbf{S}; \mathbf{0}], \mathbf{1})$. The entry $(\mathbf{m}, \alpha, sig)$ is stored so that the same signature sig is returned next time that $(\mathbf{m}, \alpha)$ is signed.

Verification. Verification of a signature $\bar{sig} = (\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2)$ on message $\bar{M} = (\bar{\mathbf{m}}, \alpha)$ returns 1 if $[\mathbf{A}|\mathbf{B}|\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}\mathbf{G}|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha)$, if $\bar{M} \in \bar{\mathcal{M}}$, and if $\bar{sig} \in \{(\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2) \in \bar{\mathcal{C}} \times \mathcal{R}_q^{3+2m} \times \mathcal{R}_q : \|\bar{\mathbf{S}}\| \leq \bar{N}_s \ \wedge \ \|\bar{\mathbf{S}}\|_\infty \leq \bar{N}_{s,\infty} \ \wedge \ \|\bar{\mathbf{c}}_2\| \leq \bar{C}\}$. Otherwise, it returns 0.

Correctness of the rS scheme follows from the following choices of the parameters. Lemma 4 guarantees that with probability at least $\frac{1}{2}$ it holds $s_1(\mathbf{X}) < \frac{\sigma_t}{\sqrt{\pi}}\sqrt{n} \cdot (\sqrt{2} + \sqrt{m} + \log(n))$, as $\mathbf{X}$ is sampled from $\mathcal{D}^{2\times m}_{\mathcal{R}_3,\sigma_t}$. Therefore, if we set the standard deviation of the Gaussian from which the signatures are sampled as $\sigma = q^{1/m}\frac{\sigma_t}{\sqrt{\pi}}\sqrt{n} \cdot (\sqrt{2} + \sqrt{m} + \log(n))$ we are able to sample from $D^{\perp}_{[\mathbf{A}|\mathbf{A}\mathbf{R}+\mathbf{G}],\mathcal{H}(\alpha),\sigma}$ thanks to Theorem 1. By Lemma 5 we are also able to sample $\mathbf{S}$ from $D^{\perp}_{[\mathbf{A}|\mathbf{A}\mathbf{R}+\mathbf{G}|\mathbf{U}+\mathbf{m}\mathbf{G}],\mathcal{H}(\alpha),\sigma}$. Therefore, Lemma 3 guarantees that the norms of a honestly generated signature can be bounded as $\|\mathbf{S}\| \leq 1.05\sigma\sqrt{n(2m+2)} = N_s < \bar{N}_s$ and $\|\mathbf{S}\|_\infty \leq 8\sigma < \bar{N}_{s,\infty}$ with high probability. Finally, any message in $\mathcal{M}_{K_m}$ is also in $\bar{\mathcal{M}}_{K_m}$ by construction. Observe that $\bar{C} \geq 1$, hence the verification algorithm accepts any signature generated by Sign.

5.3 Unforgeability

We prove the g-unforgeability of our rS scheme under Assumption 3 described below. Assumption 3 is very similar to the g-unforgeability experiment itself,

but, similarly to what we did for the hiding property of the rC scheme, we gain trust in the assumption by introducing a selective variant in Assumption 4 that we show to be implied by the Ring-LWE and Ring-SIS assumptions. A complexity leveraging argument can be used to show that Assumption 3 holds when Assumption 4 holds.

Basically, Assumption 3 states that it should be hard to find a short vector in some coset $\mathcal{L}^\perp(\mathbf{M}) + \mathbf{c}_2\mathcal{H}(\alpha)$ where $\mathbf{M} = [\mathbf{A}|\mathbf{B}|\mathbf{c}_1\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]$ (for some $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{m}$ chosen by the solver) without knowing a trapdoor for $\mathbf{M}$.

Assumption 3 *Consider the following game between an adversary* A *and a challenger for fixed $m \in \mathbb{N}$ and distribution D:*

1. *The challenger chooses $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$, $\mathbf{C} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$, and $\mathbf{X} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$. It sets $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$ and $\mathbf{B} = \mathbf{A}\mathbf{X} + \mathbf{G}$, where $\mathbf{G} = [1\ \lceil q^{1/m}\rceil\ \ldots\ \lceil q^{(m-1)/m}\rceil]$.*
2. *The challenger runs* A *on input $[\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}]$, giving it access to a random oracle $\mathcal{H} : \{0,1\}^* \to \mathcal{R}_q$ and an oracle $\mathcal{O}_S$ that on input $\mathbf{m} \in \mathcal{U}$ and a string $\alpha \in \{0,1\}^*$ outputs a small vector $\binom{\mathbf{S}}{\mathbf{1}}$ in the coset $\mathcal{L}^\perp([\mathbf{A}|\mathbf{B}|\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]) + \mathcal{H}(\alpha)$ such that $\|\mathbf{S}\| \leq N_S$.*
3. *Algorithm* A *outputs $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$, $\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2 \in \bar{\mathcal{C}}$, and a vector $\bar{\mathbf{S}}$. Algorithm* A *wins the game if $(\mathbf{c}_1, \mathbf{S}, \mathbf{c}_2) \in \bar{\Sigma}$, $\mathbf{m} \in \bar{\mathcal{U}}$, such that $\mathbf{S}$ is a short vector of the coset $\mathcal{L}^\perp([\mathbf{A}|\mathbf{B}|\bar{\mathbf{C}}|\mathbf{1}]) + \mathbf{c}_2\mathcal{H}(\alpha))$ and $\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}$ was not queried to the $\mathcal{O}_S$ oracle.*

The assumption states that no PPT algorithm A *can win the game with non-negligible probability.*

Assumption 4 (Selective variant of Assumption 3.) *Consider the game of Assumption 3, but where step 1 is preceded with a step where* A*, on input only the security parameter λ, outputs the message $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$, and in step 3 outputs the remaining items $\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2 \in \bar{\mathcal{C}}$, and $\bar{\mathbf{S}}$. The assumption states that no PPT adversary can win this previous game with non-negligible advantage.*

In the following theorem, we show that Assumption 4 is implied by the Ring-SIS and Ring-LWE assumptions.

Theorem 7 (Hardness of Assumption 4). *Let* A *be a probabilistic algorithm that breaks Assumption 4 in time t with probability ϵ_A. Then there exists a probabilistic algorithm* B *that either breaks Ring-LWE$_{m,\mathcal{D}_\sigma}$ in time t with probability ϵ_A or Ring-SIS$_{3+m,q,\beta_s}$ in time t with probability $\epsilon_B \geq (\epsilon_A - \epsilon_{\mathrm{LWE}})/(2 \cdot |\bar{\mathcal{C}}|)$, where ϵ_{LWE} is the probability of breaking the Ring-LWE problem over $\mathcal{R}_q$ in time t, in the Random Oracle Model.*

Proof. We construct the algorithm B as follows. Let $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$ be the message output by A at the beginning of the game. Algorithm B is given a vector $\mathbf{A}' = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{m+1}] \in \mathcal{R}_q^{1\times(2+m)}$ as Ring-SIS challenge. To solve Ring-SIS it should find a short vector $\mathbf{Y} \in \mathcal{R}_q^{3+m}$ such that $[\mathbf{A}'|\mathbf{1}]\mathbf{Y} = \mathbf{0}$. First, B constructs $\mathbf{A} = [\mathbf{a}_{m+1}|\mathbf{1}] \in \mathcal{R}_q^{1\times 2}$ and $\mathbf{B} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathcal{R}_q^{1\times m}$, and samples $\mathbf{R}$ from $\mathcal{D}_{\mathcal{R}_3,\sigma_t}^{2\times m}$. It guesses $\bar{\mathbf{c}}_1 \xleftarrow{\$} \bar{\mathcal{C}}$ as part of the solution of Assumption 4 that A will output

in step 3. Then algorithm B constructs the public parameter as $[\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}] = [\mathbf{A}|\mathbf{B}|\mathbf{AR} - \bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}\mathbf{G}|\mathbf{1}]$. Finally, it sends $[\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}]$ to A and handles its hash and oracle queries as follows.

Hash queries. When A makes a query $\mathcal{H}(\alpha)$, B returns its previous response if α was already queried, otherwise it programs $\mathcal{H}(\alpha)$ as follows. It samples $\mathbf{S} = [\mathbf{S}_1; \mathbf{S}_2; \mathbf{S}_3]$ from $D_{q,\sigma}^{2+2m}$, and programs $\mathcal{H}(\alpha) = [\mathbf{A}|\mathbf{B}|\mathbf{AR}]\mathbf{S}$. It stores $(\alpha, \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ and returns $\mathcal{H}(\alpha)$ to A.

Oracle queries. When A makes a query to $\mathcal{O}_S$ with input $M = (\mathbf{m}, \alpha)$, B first checks that $\mathbf{m} \in \mathcal{U}$. It then proceeds as follows:

- If $\mathbf{m} = \bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}}$, B simulates a hash query $\mathcal{O}_{\mathcal{H}}(\alpha)$ as described above and reads the corresponding $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_3$ from the list. Then it returns $sig = (\mathbf{1}, [\mathbf{S}_1; \mathbf{S}_2; \mathbf{S}_3; \mathbf{0}], \mathbf{1})$. Remark that $\bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}}$ might not be in $\mathcal{U}$. If that is the case, this part of the simulation never happens.
- If $\mathbf{m} \neq \bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}}$, B queries $\mathcal{H}(\alpha)$ to $\mathcal{O}_{\mathcal{H}}$ and samples $\mathbf{S}$ from $D^{\perp}_{[\mathbf{A}|\mathbf{B}|\mathbf{AR}+(\mathbf{m}-\bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}})\mathbf{G}],\mathcal{H}(\alpha),\sigma}$ using $\mathbf{R}$ as trapdoor and it returns $sig = (\mathbf{1}, [\mathbf{S}; \mathbf{0}], \mathbf{1})$. To guarantee that the sampling is possible, we need to check that $\mathbf{m} - \bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}} = \bar{\mathbf{c}}_1^{-1}(\bar{\mathbf{c}}_1\mathbf{m} - \bar{\mathbf{m}})$ is invertible. This is true by Lemma 1 if the numerator has infinity norm less than $2 \cdot 2^{K_m - 1} + p < \sqrt{q/2}$ (as the denominator is invertible). This holds for the choice of p in Table 7.3.

B is computationally indistinguishable from the challenger in Assumption 3 under Ring-LWE. This reduction is quite standard and can be found in Appendix E.

Upon receiving a valid solution $\big((\bar{\mathbf{m}}, \alpha'), (\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2)\big)$ from A, B aborts if $\bar{\mathbf{c}}_1$ is not the value that it guessed before. Otherwise, substituting $\mathbf{C} = \mathbf{AR} - \bar{\mathbf{c}}_1^{-1}\bar{\mathbf{m}}\mathbf{G}$ in $[\mathbf{A}|\mathbf{B}|\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}\mathbf{G}|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha')$ yields:

$$[\mathbf{A}|\mathbf{B}|\mathbf{AR}\bar{\mathbf{c}}_1|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha') \tag{6}$$

as $\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}\mathbf{G} = \mathbf{AR}\bar{\mathbf{c}}_1 - \bar{\mathbf{c}}_1\frac{\bar{\mathbf{m}}}{\bar{\mathbf{c}}_1}\mathbf{G} + \bar{\mathbf{m}}\mathbf{G} = \mathbf{AR}\bar{\mathbf{c}}_1$.
Now, algorithm B simulates a query $\mathcal{O}_{\mathcal{H}}(\alpha')$ and recovers $(\alpha, \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ from the list. For $\mathbf{T} = [\mathbf{T}_1; \mathbf{T}_2] = [\mathbf{S}_1 + \mathbf{RS}_3; \mathbf{S}_2]$ we have that $[\mathbf{A}|\mathbf{B}]\mathbf{T} = \mathcal{H}(\alpha')$. Combining this with Equation (6) and decomposing $\bar{\mathbf{S}}$ as $\bar{\mathbf{S}} = [\mathbf{S}'_1; \mathbf{S}'_2; \mathbf{S}'_3; \mathbf{s}'_4]$ yields:

$$[\mathbf{B}|\mathbf{A}] \begin{pmatrix} \mathbf{S}'_1 + \bar{\mathbf{c}}_1\mathbf{RS}'_3 - \bar{\mathbf{c}}_2\mathbf{T}_1 + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'_4 \end{pmatrix} \\ \mathbf{S}'_2 - \bar{\mathbf{c}}_2\mathbf{T}_2 \end{pmatrix} = \mathbf{0} \ . \tag{7}$$

Denote the vector on the left-hand side as $\mathbf{V} = [\mathbf{V}_1; \mathbf{V}_2]$.

If $\mathbf{V} \neq \mathbf{0}_{(m+3)\times 1}$, then we have that $[\mathbf{B}|\mathbf{A}]\mathbf{V} = [\mathbf{A}'|\mathbf{1}]\mathbf{V} = \mathbf{0}$, so that B obtained a solution for Ring-SIS with norm bounded by β_s. Indeed, the norm of $\mathbf{V}$ is bounded by the norm of $\mathbf{V}_1$. By the triangular inequality, Lemma 2 and Lemma 4 the following bound holds:

$$\begin{aligned} \|\mathbf{V}_1\|^2 &\leq \|\mathbf{S}'_1\|^2 + (\mathsf{s}_1(\mathbf{R})\|\bar{\mathbf{c}}_1\|\|\bar{\mathbf{S}}_3\|)^2 n + (\|\bar{\mathbf{c}}_2\|\mathsf{s}_1(\mathbf{S}_1))^2 + (\mathsf{s}_1(\mathbf{R})\|\mathbf{S}_3\|\|\bar{\mathbf{c}}_2\|)^2 n + \|\mathbf{s}'_4\|^2 \\ &\leq 2n\bar{N}_{s,\infty} + \frac{\sigma_t^2}{\pi} n^2 (\sqrt{2} + \sqrt{m} + \log n)^2 (2 \cdot 2\sqrt{2^{K_m - 2} - 1})^2 nm\bar{N}_{s,\infty} + \\ &\quad + \bar{C}^2 \frac{\sigma^2}{\pi} n(1 + \sqrt{2} + \log n)^2 + \frac{\sigma_t^2}{\pi} n(\sqrt{2} + \sqrt{m} + \log n)^2 (1.05\sigma\sqrt{nm})^2 \bar{C}^2 n \end{aligned}$$

$=: B$

where we assume it holds that $n\|\mathbf{S}'_3\|_\infty\|\bar{\mathbf{c}}_1\|_\infty \leq \frac{q-1}{2}$ and $n\|\mathbf{S}_3\|_\infty\|\bar{\mathbf{c}}_2\|_\infty \leq \frac{q-1}{2}$ (this holds for the choice of parameters in Table 7.3). We also used the fact that the degree of $\mathbf{c} \in \bar{\mathcal{C}}$ is strictly less than $n/4$, thus the number of its nonzero coefficients is $2^{K_m-2}-1$. Hence, $\|\mathbf{V}\| \leq 2^{\lceil \log_2(\sqrt{B})\rceil}$.

If $\mathbf{V} = \mathbf{0}_{(m+3)\times 1}$, then we have in particular that $\mathbf{V}_1 = \mathbf{0}_{2\times 1}$ and $\mathbf{V}_2 = \mathbf{0}_{m\times 1}$, from which it follows that $\mathbf{S}'_1 + \bar{\mathbf{c}}_1\mathbf{R}\mathbf{S}'_3 + \binom{\mathbf{0}}{\mathbf{s}'_4} = \bar{\mathbf{c}}_2\mathbf{T}_1$ and $\mathbf{S}'_2 = \bar{\mathbf{c}}_2\mathbf{T}_2$. Multiplying both sides of both equations with $\bar{\mathbf{c}}_2^{-1}$ yields $\mathbf{T}_1 = \bar{\mathbf{c}}_2^{-1}(\mathbf{S}'_1 + \bar{\mathbf{c}}_1\mathbf{R}\mathbf{S}'_3 + \binom{\mathbf{0}}{\mathbf{s}'_4})$ and $\mathbf{T}_2 = \bar{\mathbf{c}}_2^{-1}\mathbf{S}'_2$. Recall that $\|\bar{\mathbf{c}}_2\| < q/2$ and q is a prime, $q \equiv 5 \bmod 8$, therefore $\mathbf{c}'$ is invertible by Lemma 1. Now, consider the deterministic function $h : \mathcal{R}_q^{2+m} \to \mathcal{R}_q$ where $h(\mathbf{Y}) = [\mathbf{A}|\mathbf{B}]\mathbf{Y}$. By Lemma 6, for a randomly chosen $\mathbf{Y}$ there exists with probability at least $1 - 2^{-\lambda}$ another vector $\mathbf{Y}' \neq \mathbf{Y}$ such that $h(\mathbf{Y}) = h(\mathbf{Y}')$. The parameter λ should be such that $|\mathcal{R}_q^{m+2}| \geq 2^\lambda |\mathcal{R}_q|$, thus $\lambda \leq \log q \cdot n(m+1)$. Moreover, A's view is (computationally) independent of B's choice for $\mathbf{T}$, because only its image $h(\mathbf{T}) = [\mathbf{A}|\mathbf{B}]\mathbf{T} = \mathcal{H}(\alpha')$ was output by the hash function and because $\mathbf{T}$ was never used to simulate a query to $\mathcal{O}_S$. Indeed, the only query that would involve $\mathbf{T}$ in the simulation is $\mathcal{O}_S((\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}, \alpha'))$:

- if $\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1} \in \mathcal{U}$, A never queried for $(\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}, \alpha')$ (otherwise the output of A would not be a valid solution);
- if $\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1} \notin \mathcal{U}$ this query never happened, as $(\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}, \alpha')$ would not be accepted as input by $\mathcal{O}_S$.

Hence, the probability that A outputs a $\mathbf{T}' = [\bar{\mathbf{c}}_2^{-1}(\mathbf{S}'_1 + \bar{\mathbf{c}}_1\mathbf{R}\mathbf{S}'_3)\ ;\ \bar{\mathbf{c}}_2^{-1}\mathbf{S}'_2]$ such that $\mathbf{T}' = \mathbf{T}$ is at most $\frac{1}{2}$.

Therefore, B outputs a nonzero solution of Ring-SIS with probability $\epsilon_B \geq \frac{\epsilon_A - \epsilon_{\mathrm{LWE}}}{2\cdot|\bar{\mathcal{C}}|}$ in time t, where ϵ_{LWE} is the probability of breaking the Ring-LWE problem over $\mathcal{R}_q$ in time t. □

The following theorem states that breaking Assumption 4 implies breaking Assumption 3. It follows from a straightforward complexity leveraging argument by guessing the polynomial $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$.

Theorem 8. *Let* A *be a probabilistic algorithm that breaks Assumption 3 in time t with probability ϵ_A. Then, there exists a probabilistic algorithm* B *that breaks Assumption 4 in time t with probability $\epsilon_B \leq \epsilon_A/|\bar{\mathcal{U}}|$ in the Random Oracle Model.*

We define the g-uf-cma security of the relaxed signature scheme with respect to the message relaxation function

$$g(\mathbf{m}, \alpha) \ = \ \{(\bar{\mathbf{m}}, \alpha) \ : \ \bar{\mathbf{m}} \in f(\mathbf{m})\} \ ,$$

where the function f is as defined in Equation (2). A valid forgery is a signature $(\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2)$ on some message $(\bar{\mathbf{m}}, \alpha')$ such that the adversary never saw a signature on $(\mathbf{m}, \alpha')$ for any $\mathbf{m}$ such that $\bar{\mathbf{m}} \in f(\mathbf{m})$. The unforgeability of the signature scheme follows directly from Assumption 3.

Theorem 9. *An algorithm* A *that breaks the g-uf-cma unforgeability of the relaxed signature scheme in time t and probability* ϵ_A *can break the Assumption 3 in time t with probability* ϵ_A *in the Random Oracle Model.*

A valid forgery can be used to break Assumption 3 because unforgeability is defined w.r.t. a function g. This guarantees that $\bar{\mathbf{m}}$ and $\bar{\mathbf{c}}_1$ output by A are such that $\bar{\mathbf{m}}\bar{\mathbf{c}}_1^{-1}$ was not queried to $\mathcal{O}_S$ as specified by the assumption.

6 Relaxed Proofs of Signatures on Committed Messages

Our three primitives can be composed together to prove knowledge of a signature $\mathbf{S}$ on a secret $\mathbf{m}$ w.r.t. a public bit-string α. To prove knowledge of both $\mathbf{S}$ and $\mathbf{m}$, the protocol exploits the relaxed commitment scheme defined in Section 4.3. The commitment is needed both for technical and practical reasons. Indeed, it allows to prove knowledge of the signature and the secret part of the message in two separate equations. On one hand, this gives a better bound on the extracted message, as rejection sampling can be performed separately on the two equations. On the other hand, this allows to prove knowledge of a set of signatures $\{\mathbf{S}_i\}_{i=1,\ldots,\ell}$ on messages $(\mathbf{m}, \alpha_i)$ for $i = 1\ldots, \ell$, i.e. on message pairs composed by the same secret $\mathbf{m}$ and by different public bit-strings α_i.

We start presenting the proof for a single pair message-signature. The generalization to the multiple-signatures case is shown at the end of this section. Let $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ be the public vectors of the signature scheme. Let $\mathcal{H}_c$ be a hash function as in Section 3.2. Given α and the public parameters of the signature, $\mathcal{P}$ wants to prove that she owns some "small" $(\mathbf{m}, (\mathbf{c}_1, \mathbf{S}, \mathbf{c}_2))$ such that $[\mathbf{A}|\mathbf{B}|\mathbf{c}_1\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]\mathbf{S} = \mathbf{c}_2\mathcal{H}(\alpha)$. To construct a relaxed NIZK proof (cfr. Section 3.2), we rewrite the characterizing equation as it follows. Let $(\mathbf{1}, \mathbf{S}, \mathbf{1})$ be a honestly-generated signature on $(\mathbf{m}, \alpha)$, i.e.

$$[\mathbf{A}|\mathbf{B}|\mathbf{1}\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]\mathbf{S} \;.= \mathbf{1}\mathcal{H}(\alpha) \tag{8}$$

We generate a commitment $\mathbf{F} = \mathbf{b}^{-1}(\mathbf{C} + \mathbf{m}\mathbf{G} + \mathbf{E})$ to $\mathbf{m}$ and we substitute $\mathbf{C} + \mathbf{m}\mathbf{G} = \mathbf{F}\mathbf{b} - \mathbf{E}$ in Equation (8). Rearranging the terms it follows that with this relaxed NIZK proof $\mathcal{P}$ shows she owns some "small" $(\bar{\mathbf{S}}_c, \bar{\mathbf{c}}_1, \bar{\mathbf{S}}_s, \bar{\mathbf{c}}_2)$ satisfying:

$$\text{(I)}\ \underbrace{\left[-\mathbf{G}^T\ \mathbf{F}^T\ -\mathbb{I}_m\right]}_{=\mathbf{A}_c}\underbrace{\begin{pmatrix}\bar{\mathbf{m}}\\ \bar{\mathbf{b}}\\ \bar{\mathbf{E}}^T\end{pmatrix}}_{=\bar{\mathbf{S}}_c} = \bar{\mathbf{c}}_1\mathbf{C}^T\ ,\ \text{(II)}\ \underbrace{[\mathbf{A}|\mathbf{B}|\mathbf{F}|\mathbf{1}]}_{=\mathbf{A}_s}\underbrace{\begin{pmatrix}\bar{\mathbf{S}}_1\\ \bar{\mathbf{S}}_2\\ \bar{\mathbf{S}}_3\\ \bar{\mathbf{s}}_4\end{pmatrix}}_{=\bar{\mathbf{S}}_s} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha). \tag{9}$$

To define the relations for the relaxed Σ-protocol, we need first to bound the norm of the secrets. The vector $\mathbf{S}_c = [\mathbf{m}; \mathbf{b}; \mathbf{E}^T]$ is composed by $1 + m$ polynomials in $\mathcal{R}_3$ and one in $\mathcal{R}_3^{(2^{K_m})}$ of degree $n/2$: its norm is bounded by $N_1 = \sqrt{(1+m)n + 2^{2^{K_m-1}}}$. For the norm of $\mathbf{S}_s = [\mathbf{S}_1; \mathbf{S}_2; \mathbf{S}_3; -\mathbf{E}\mathbf{S}_3]$, the first $2 + 2m$ components have norm bounded by $1.05\sigma\sqrt{n}$, while the norm of the last com-

ponent can be bound using the properties of the singular value:

$$\|\mathbf{E}\mathbf{S}_3\| \leq \|\mathbf{E}\| \cdot \mathsf{s}_1(\mathbf{S}_3) \leq \sqrt{nm} \cdot \frac{\sigma}{\sqrt{\pi}}\sqrt{n}(1+\sqrt{m}+\log n) \ .$$

Setting $N_2 = \sqrt{(2+2m)(1.05\sigma\sqrt{n})^2 + (n\frac{\sigma\sqrt{m}}{\sqrt{\pi}}(1+\sqrt{m}+\log n))^2}$ allows us to define the following relations:

$$\begin{aligned} R = \{&((\mathbf{A}_c, \mathbf{C}^T, \mathbf{A}_s, \mathcal{H}(\alpha)), (\mathbf{S}_c, \mathbf{S}_s, \mathbf{1})) \ : \ \mathbf{m} \in \mathcal{U} \\ &(\mathbf{S}_c, \mathbf{1}) \text{ satisfies relation (I) and } \|\mathbf{S}_c\| \leq N_1 \ , \ \|\mathbf{S}_c\|_\infty < (q-1)/(2n) \\ &(\mathbf{S}_s, \mathbf{1}) \text{ satisfies relation (II) and } \|\mathbf{S}_s\| \leq N_2 \ , \ \|\mathbf{S}_s\|_\infty < (q-1)/(2n)\} \end{aligned} \tag{10}$$

$$\begin{aligned} \bar{R} = \{&((\mathbf{A}_c, \mathbf{C}^T, \mathbf{A}_s, \mathcal{H}(\alpha)), (\bar{\mathbf{S}}_c, \bar{\mathbf{S}}_s, \bar{\mathbf{c}})) \ : \ \bar{\mathbf{m}} \in \bar{\mathcal{U}} \\ &(\bar{\mathbf{S}}_c, \bar{\mathbf{c}}) \text{ satisfy relation (I) and : } \|\bar{\mathbf{S}}_c\| \leq \bar{N}_1 \ , \ \|\bar{\mathbf{S}}_c\|_\infty \leq \bar{N}_{1,\infty} \\ &(\bar{\mathbf{S}}_s, \bar{\mathbf{c}}) \text{ satisfy relation (II) and : } \|\bar{\mathbf{S}}_s\| \leq \bar{N}_2 \ , \ \|\bar{\mathbf{S}}_s\|_\infty \leq \bar{N}_{2,\infty}\} \end{aligned} \tag{11}$$

for some constants $\bar{N}_1$, $\bar{N}_{1,\infty}$, $\bar{N}_2$, $\bar{N}_{2,\infty}$. Let $\mathcal{P}$ and $\mathcal{V}$ be the prover and verifier defined in Section 3.2 w.r.t. the relations (10) and (11). To prove that $\mathbf{m}$ is in $\bar{\mathcal{U}}$ (i.e., $\mathbf{m}$ is an element of $\mathcal{R}_{2p+1}^{(2^{K_m})}$ with degree $3n/4$) set the challenge spaces $\mathcal{C}$ and $\bar{\mathcal{C}}$ to be as in Section 4.2. To have better bounds on the message and on the opening information, $\mathcal{P}$ does rejection sampling separately for $\mathbf{S}_s$ and $\mathbf{S}_c$. Hence, $\mathcal{P}$ samples $\mathbf{Y}_1 \xleftarrow{\$} \mathcal{D}_{\mathcal{Y}_{3n/4} \times \mathcal{R}_q^{1+m}, \sigma_1}$ and $\mathbf{Y}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q, \sigma_2}^{2m+3}$ and sends as commitments $(\mathbf{A}_c\mathbf{Y}_1, \mathbf{A}_s\mathbf{Y}_2)$. Upon receiving the challenge $\mathbf{c}$ from the verifier, the prover sets $\mathbf{Z}_1 = \mathbf{Y}_1 + \mathbf{c}\mathbf{S}_c$ and $\mathbf{Z}_2 = \mathbf{Y}_2 + \mathbf{c}\mathbf{S}_s$, and does rejection sampling separately on them. If $\sigma_1 = 12T_1$ and $\sigma_2 = 12T_2$ (where T_1, T_2, are upper bounds on the norm of $\mathbf{c}\mathbf{S}_c$, $\mathbf{c}\mathbf{S}_s$ respectively) the probability that $\mathcal{P}$ outputs $(\mathbf{Z}_1, \mathbf{Z}_2)$ is greater than $1/8$ (cfr. Lemma 4.3, 4.4, 4.5 in [Lyu12] and Section 3.2). To compute the values T_1, T_2 we observe that a challenge $\mathbf{c} \in \mathcal{C}$ has norm bound by $\|\mathbf{c}\| \leq \sqrt{2^{2^{K_m-2}-1}}$, hence using Lemma 2 we can set $T_i = N_i\sqrt{n2^{2^{K_m-2}-1}}$ for $i = 1, 2$.

To guarantee special soundness, we set $\bar{N}_1 = 2.1\sigma_1\sqrt{n(2+m)}$, $\bar{N}_2 = 2.1\sigma_2 \sqrt{n(2+2m)}$ and $\bar{N}_{1,\infty} = 16\sigma_1$, $\bar{N}_{2,\infty} = 16\sigma_2$ as in Section 3.2. Setting $p = \bar{N}_{1,\infty}$ guarantees that $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$.

The cardinality of $\mathcal{C}$ is $|\mathcal{C}| = 3^{2^{K_m-2}-1} = 3^{15}$ when $K_m = 6$, hence the proof has to be repeated 6 times to have negligible soundness error.

Theorem 10. *Given N_s as in Section 5.2 and $\bar{N}_s = 8.82(3+2m)\sigma_1\sigma_2 n\sqrt{nm}$, $\bar{N}_{s,\infty} = 512\sigma_1\sigma_2 n$, $\bar{C} = 4.2\sigma_1 n\sqrt{2^{K_m-2}-1}$ and $p = 16\sigma_1$, the protocol $(\mathcal{P}, \mathcal{V})$ is a relaxed Σ-protocol for the following pair of relations:*

$$\begin{aligned} R = \{ &((\mathbf{A}_s, \mathcal{H}(\alpha)), (\mathbf{m}, (\mathbf{1}, \mathbf{S}, \mathbf{1}))) \ : \ \mathbf{m} \in \mathcal{U}, \\ &[\mathbf{A}|\mathbf{B}|\mathbf{1}\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]\mathbf{S} = \mathbf{1}\mathcal{H}(\alpha) \text{ and } \|\mathbf{S}\| \leq N_s\} \\ \bar{R} = \{ &((\mathbf{A}_s, \mathcal{H}(\alpha)), (\bar{\mathbf{m}}, (\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2))) \ : \ \bar{\mathbf{m}} \in \bar{\mathcal{U}}, \ \bar{\mathbf{c}}_1 \in \bar{\mathcal{C}}, \ \|\bar{\mathbf{c}}_2\| \leq \bar{C} \\ &[\mathbf{A}|\mathbf{B}|\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}\mathbf{G}|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha) \text{ and } \|\bar{\mathbf{S}}\| \leq \bar{N}_s, \ \|\bar{\mathbf{S}}\|_\infty \leq \bar{N}_{s,\infty}\} \end{aligned}$$

under Ring-LWE$_{\mathcal{R}_3}$ with the uniform distribution.

Proof. Correctness is trivial.

The HVZK of this protocol can be proved by constructing a simulator very similar to the one in Theorem 2. The simulator is indistinguishable from uniform thanks to the anonymity of the commitment scheme and to rejection sampling.

Finally, we prove special soundness. A knowledge extractor $(\mathsf{E}_1^s, \mathsf{E}_2^s)$ rewinds Π to obtain the vectors $[\bar{\mathbf{b}}; \bar{\mathbf{m}}; \bar{\mathbf{E}}]$, $[\bar{\mathbf{S}}_1; \bar{\mathbf{S}}_2; \bar{\mathbf{S}}_3; \bar{\mathbf{s}}_4]$ and the polynomial $\bar{\mathbf{c}}$ that satisfy equations (I) and (II) in (9). Multiplying equation (II) times $\bar{\mathbf{b}}$ and plugging in $\bar{\mathbf{b}}\mathbf{F} = \bar{\mathbf{c}}\mathbf{C} + \bar{\mathbf{m}}\mathbf{G} + \bar{\mathbf{E}}$ yields:

$$\mathbf{A}(\bar{\mathbf{b}}\bar{\mathbf{S}}_1) + \mathbf{B}(\bar{\mathbf{b}}\bar{\mathbf{S}}_2) + [\bar{\mathbf{c}}\mathbf{C} - \bar{\mathbf{m}}\mathbf{G}](\bar{\mathbf{S}}_3) + (\bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3) = \bar{\mathbf{c}}\bar{\mathbf{b}}\mathcal{H}(\alpha) \qquad (12)$$

The vector $\bar{\mathbf{S}} = [\bar{\mathbf{b}}\bar{\mathbf{S}}_1; \bar{\mathbf{b}}\bar{\mathbf{S}}_2; \bar{\mathbf{S}}_3; \bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3]$ has norm bounded by the norm of $\bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3$. The element with the largest norm is $\bar{\mathbf{E}}\bar{\mathbf{S}}_3$ and applying Lemma 2 we have $\|\bar{\mathbf{E}}\bar{\mathbf{S}}_3\| \leq 2.1\sigma_1\sqrt{n} \cdot 2.1\sigma_2\sqrt{n} \cdot \sqrt{nm}$, where the inequality holds as by Lemma 3 we can bound the product of the infinity norms as $\|\bar{\mathbf{E}}\|_\infty \|\bar{\mathbf{S}}_3\|_\infty n < 16\sigma_1 \cdot 16\sigma_2 n$ that is less than $q/2$ for our choice of parameters. Hence, setting $\bar{N}_s = (3+2m) \cdot 2(2.1\sigma_1\sqrt{n} \cdot 2.1\sigma_2\sqrt{n} \cdot \sqrt{nm})$ we have that $\|\bar{\mathbf{S}}\| \leq \bar{N}_s$ and, again from Lemma 2 $\|\bar{\mathbf{S}}\|_\infty \leq 2 \cdot 16\sigma_1 \cdot 16\sigma_2 n =: \bar{N}_{s,\infty}$. Observe that $\bar{N}_{s,\infty} > 8\sigma_2 > 8\sigma$, hence the correctness of the signature scheme is guaranteed (see Section 5.2). Moreover, $\bar{\mathbf{c}} \in \bar{\mathcal{C}}$, and, applying again Lemma 2, $\|\bar{\mathbf{c}}\bar{\mathbf{b}}\| \leq \|\bar{\mathbf{c}}\|\|\bar{\mathbf{b}}\|\sqrt{n} \leq 2\sqrt{2^{K_m-2}-1} \cdot 2.1\sigma_1 n =: \bar{C}$. Hence $(\bar{\mathbf{c}}, \bar{\mathbf{S}}, \bar{\mathbf{c}}\bar{\mathbf{b}}) \in \bar{\Sigma}$, i.e. the extractor outputs a valid signature. From point 2 in Lemma 3 the infinity norm of the extracted user's secret key $\bar{\mathbf{m}}$ is less than $16\sigma_1 =: p$. □

The protocol is made non interactive via the construction presented in Theorem 13 with the Lamport signature as OTS (Appendix B).

A proof of knowledge of ℓ signatures $\mathbf{S}_i$ generated by signer i on ℓ messages $(\mathbf{m}, \alpha_i)$ is constructed by combining ℓ of the previous proofs in parallel. Assume that the parameters of the rC and rS schemes are shared among all signers. This means that the verification key of signer j is $[\mathbf{A}_j|\mathbf{B}_j|\mathbf{C}]$ for the same $\mathbf{C}$. Hence, the prover can generate a commitment $\mathbf{F}$ to $\mathbf{m}$ using $\mathbf{C}$ as public matrix, and generate a proof Π_i that she knows a secret $\bar{\mathbf{S}}_c$ that satisfies relation (I) in (9) and $\bar{\mathbf{S}}_{s,i}$, $\bar{\mathbf{c}}$ that satisfy $[\mathbf{A}_i|\mathbf{B}_i|\mathbf{F}|\mathbf{1}]\bar{\mathbf{S}}_{s,i} = \bar{\mathbf{c}}\mathcal{H}(\alpha_i)$ for $i = 1, \ldots, \ell$. The relaxed binding property of the commitment guarantees that the hidden part of the message $\mathbf{m}$ is the same in all proofs.

7 Compact Anonymous Attribute Tokens from Lattices

Anonymous attribute tokens [CNR12] can be seen as simplified anonymous credentials, allowing users to obtain a credential from an issuer that contains a list of attributes. Users can selectively disclose subsets of these attributes to verifiers in such a way that not even the verifier and the issuer together can link different presentations by the same user. In this section, we focus on AAT schemes without opening (AAT-O), i.e., without a trusted opener who can deanonymize presentation tokens. In Appendix G, we also provide a construction of an AAT scheme with opening (AAT+O) using verifiable encryption [LN17], which immediately gives rise to a group signature scheme.

7.1 Definition of AAT-O Schemes

We summarize the definition of AAT-O schemes here; the more formal and detailed definition of AAT-O schemes can be found in Appendix A.3.

An issuer generates a public key *ipk* and corresponding secret key *isk* by running $\mathsf{IKGen}(par)$. To issue a credential for attributes $(\alpha_i)_{i=1}^{\ell}$, the issuer choses an unused user identity *id* and runs $\mathsf{Issue}(isk, id, (\alpha_i)_{i=1}^{\ell})$ and hands *id* and the resulting credential *cred* to the user. A user creates a presentation token *pt* revealing a subset of attributes $(\alpha_i)_{i\in R}$, $R \subseteq \{1, \ldots, \ell\}$, from a credential while authenticating a message M by running $\mathsf{Present}(ipk, cred, R, M)$. A verifier can verify a presentation token by running $\mathsf{Verify}(ipk, R, (\alpha_i)_{i\in R}, M, pt)$.

Unforgeability requires that no PPT adversary with access to an issuance oracle and an oracle that generates presentation tokens by honest users can create a presentation token revealing a set of attributes that was never issued in one credential by the issuance oracle, nor presented in that combination and for the given message by the presentation oracle.

Anonymity requires that no PPT adversary can distinguish between two presentation tokens for the same attributes and message, but derived from different credentials provided by the adversary.

7.2 Compact AATs without Opening from Lattices

From the relaxed primitives that we introduced, it is possible to construct an AAT-O scheme with compact presentation tokens. Parameters for the commitment scheme are generated from the signature parameters *spar* using the algorithm $\mathsf{DerivePar}_c$ that, on input *spar*, sets the commitment public matrix $\mathbf{C}$ to be the third block of the signature public key $vk = [\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}]$ and computes $N'_{c,2}, N'_{c,\infty}$.

System Parameter Generation. The system parameters are the signature parameters *spar* from Section 5.2. Then it runs $cpar \leftarrow \mathsf{DerivePar}_c(spar)$ and outputs $par = (spar, cpar)$.

Issuer Key Generation. The issuer runs the signing key generation $\mathsf{SignKeyGen}$ to obtain $isk = \mathbf{X}$ and the public matrix $ipk = [\mathbf{A}|\mathbf{B}|\mathbf{C}|\mathbf{1}]$.

Issuance. To issue a credential to a user for attributes $(\alpha_i)_{i=1}^{\ell}$, the issuer chooses an $id = \mathbf{m} \in \mathcal{U}$, checks that $\mathbf{m} \notin \mathcal{S}$ and computes signatures on $(\mathbf{m}, i\|att_i)$ using the Sign algorithm. The credential consists of $\mathbf{m}$, $(\alpha_i)_{i=1}^{\ell}$ together with the resulting signatures $(\mathbf{1}, [\mathbf{S}_i; \mathbf{1}], \mathbf{1})$. The issuer adds $\mathbf{m}$ to $\mathcal{S}$.

Presentation. To create a presentation token for attributes $(\alpha_i)_{i\in R}$ and message M, the user creates a commitment $\mathbf{F}$ to $\mathbf{m}$ and generates NIZK proofs Π_i that he knows signatures on the committed message and $i\|\alpha_i$ for $i \in R$, whereby he includes the message M in the Fiat-Shamir hash. The presentation token *pt* consists of the commitment $\mathbf{F}$ and the transcripts $(\Pi)_{i\in R}$.

Verification. The verifier checks the validity of $(\Pi)_{i\in R}$ w.r.t. $\mathbf{F}$ and the message M. If the tests pass, he outputs *accept*, otherwise *reject*.

Security. The security of this AAT-O follows from the security guarantees of its building blocks. Unforgeability relies on the relaxed unforgeability of the rS scheme, on the relaxed binding property of the rC scheme and on the relaxed simulation soundness of the $\mathsf{r\Sigma}$ scheme. The proof strategy is to run the adversary and extract from the forged presentation token using the Generalized Forking Lemma (Lemma 7 in Appendix C).

Theorem 11 (Unforgeability). *Assume* A *is an adversary that runs in time* t_A*, makes* q_D *random-oracle queries for credentials issued to dishonest users (if* A *queries for a credential on* $(id, (\alpha_i)_{i=1,\ldots,m})$*, we count it as* m *queries) and* q_H *queries for credential issued and presentation tokens of honest users and breaks the unforgeability of the group signature scheme with probability* ϵ_A*, then there exists an algorithm that breaks the unforgeability of the signature in time* $t_B = 32t_a(q_D + q_H)/\epsilon_A \cdot \ln(16/\epsilon_A)$ *with probability* $\epsilon_B = \epsilon_A/8$ *after asking* q_D *queries to the signing oracle in the Random Oracle Model.*

Proof (sketch). The simulator B has access to an oracle $\mathcal{O}_{sign}$ that on input $(\mathbf{m}, \alpha)$ outputs a signature on it. To win the signature unforgeability game, B runs A simulating the oracle as it follows:

Issuance to corrupt user: on input attributes $(\alpha_i)_i$, it chooses $\mathbf{m} \in \mathcal{U}$, checks whether $\mathbf{m} \in \mathcal{S}$ and queries $\mathcal{O}_{sign}$ with $(\mathbf{m}, i\|\alpha_i)$. It returns $\mathbf{m}$ and the outputs of the signing oracle and it stores $\mathbf{m}$ in $\mathcal{S}$.

Issuance to honest users: on input attributes $(\alpha_i)_i$, it selects a random *cid* and stores $((\alpha_i)_i, cid)$.

Presentation by honest users: on input attributes $(\alpha_i)_i$, it outputs a honestly generated commitment to a uniformly random $\mathbf{m} \in \mathcal{U}$ and simulates the proofs of knowledge.

Let $(pt^*, R^*, (\alpha_i^*)_{i \in R^*}, M^*)$ be the forgery output by A. The simulator rewinds A using the Generalized Forking Lemma (cf. Appendix C) and extracts an identities $\bar{\mathbf{m}}_i$ and signatures $\bar{sig}_i$ on $(\bar{\mathbf{m}}_i, i\|\alpha_i^*) \in \bar{\mathcal{M}}$. By the relaxed binding property of the commitment scheme, the identities $\bar{\mathbf{m}}_i$ are such that there exists $\mathbf{m} \in \mathcal{U}$ such that $\{\bar{\mathbf{m}}_i\}_i \subseteq f(\mathbf{m})$. Moreover, for it to be a valid forgery, there should exist at least one α_i^* that either was not part of any issued credential, or a pair of attributes α_i^*, α_j^* that were issued within different credentials. In the first case, $(\bar{\mathbf{m}}_i, i\|\alpha_i^*)$ with signature $\bar{sig}_i$ is a valid forgery. In the second case, this means that the signing algorithm signed messages $(\mathbf{m}_i, i\|\alpha_i^*)$, $(\mathbf{m}_j, j\|\alpha_j^*)$ in $\mathcal{M}_{K_m}$ for some distinct $\mathbf{m}_i$, $\mathbf{m}_j$ such that $f(\mathbf{m}_i) \cap f(\mathbf{m}_j) = \emptyset$ (by construction of $\mathcal{U}$). Then either $(\bar{\mathbf{m}}_i, i\|\alpha_i^*), \bar{sig}_i$ or $(\bar{\mathbf{m}}_j, j\|\alpha_j^*), \bar{sig}_j$ is a valid forgery. By the Generalized Forking Lemma, the success probability of B is $\epsilon_B = \epsilon_A/8$ and the running time of B is $t_B = 8\ell^2 t_a(q_D + q_H)/\epsilon_A \cdot \ln(8\ell/\epsilon_A)$. □

Anonymity is guaranteed by the zero-knowledge property of $\mathsf{r\Sigma}$ and by the hiding property of the rC scheme.

Theorem 12 (Anonymity). *If an adversary* A *running in time* t *breaks the anonymity of the group signature scheme with probability at most* ϵ*, then there*

is an adversary running in time t who breaks the anonymity of the commitment scheme with advantage at most ϵ in the Random Oracle Model.

Proof (sketch). The proof is a standard sequence of game hops. Game 0 executes the anonymity game with $b = 0$. In the first game hop the challenge oracle simulates the proofs while still generating the commitment honestly. Indistinguishability follows from the zero-knowledge of the NIZK proofs. In the second game, it substitutes the commitment to the identity $\mathbf{m}_0$ with a commitment to $\mathbf{m}_1$. Indistinguishability is guaranteed by the hiding property of the commitment. Finally, in the last game the challenge oracle outputs honestly generated proofs and a commitment to $\mathbf{m}_1$. This is exactly the anonymity game for $b = 1$. □

7.3 Parameters, Storage Requirements and a Simple Optimization

We present six different sets of parameters, depending on the level of security that is required. To compute them, we follow the general methodology from Alkim et al. [ADPS16]. This will give us a wide choice of parameters, from an optimistic choice that only guarantees classical security to a very pessimistic choice for quantum security. Ring-SIS and Ring-LWE are analyzed in their corresponding forms of SIS and LWE, as there are no known attacks that exploit the ring structure. The best algorithm to find short vectors in a lattice is the BKZ algorithm, whose latest version was published by Chen and Nguyen [CN11]. This algorithm reduces the lattice basis into blocks of size b and then calls an SVP (Shortest Vector Problem) oracle on such blocks. When computing the runtime of BKZ, we will ignore the number of calls to the oracle, as it is known to be polynomial [HPS11] and it is rather complex to compute. This makes all parameters choices significantly more conservative than needed in reality. Now, considering the SVP oracle, Alkim et al. estimated the heuristic complexity as it follows: for classical algorithms (e.g., lattice sieve algorithms) it is around $\approx 2^{0.292b}$, for quantum algorithms (e.g., sieving plus Grover's algorithm) around $\approx 2^{0.265b}$ and overall they would not go below a heuristic complexity of $\approx 2^{0.2075b}$ excluding major theory breakthroughs. To estimate the optimal block size b, we use the Hermite root factor δ: we first compute δ for the SVP instance, then we obtain b from the (optimistic estimate) Hermite root factor of the solution output by BKZ $\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{1/2(b-1)}$. We do not take into account other types of attack (e.g. [AG11,KF15]), as for our choice of parameters they would be not effective.

For each of the 3 possible scenarios, we give two sets of parameters, distinguishing whether security is based on complexity leveraging. Recall that such a technique is used in the reductions in Theorem 8 and in Theorem 4. Basing the security of the scheme on these reductions means that parameters have to compensate for the loss in tightness. Not relying on complexity leveraging means to assume that the hardness of Assumption 3 (resp. Assumption 2) implies the hardness of Assumption 4 (resp. Assumption 1) *without any tightness loss.*

			Parameters				Sizes		
Compl. Lev.	Security	δ	n	q	m	σ_t	ipk(KB)	usk(KB)	$token$(MB)
NO	classical	1.003735	2^{10}	$\sim 2^{88}$	13	4	304.128	98.24	1.5
NO	quantum	1.003488	2^{10}	$\sim 2^{88}$	14	4	323.656	103.36	1.58
NO	worst-case	1.002926	2^{10}	$\sim 2^{88}$	17	4	394.24	113.984	1.86
YES	classical	1.0005036	2^{11}	$\sim 2^{92}$	52	4	2472.96	462.272	11.15
YES	quantum	1.0004646	2^{11}	$\sim 2^{92}$	57	4	2708.48	503.232	12.19
YES	worst-case	1.0003788	2^{11}	$\sim 2^{92}$	70	4	3320.832	609.728	14.7

Table 1. Table of parameters for the AAT scheme without opening. All the values are rounded up.

Parameters that guarantee 128 bits of security are shown in Table 7.3. The third column contains the maximum value of the Hermite root factor that guarantees 128 bits of security in the different cases. As message space, we have chosen $K_m = 6$, hence $\mathcal{U} \subseteq \mathcal{R}_3^{(64)}$, and the same for the challenge space, $\mathcal{C} \subseteq \mathcal{R}_3^{(64)}$ (thus the proofs have to be repeated 6 times). In case complexity leveraging is used, the values of δ are computed taking into account the necessary compensation for the loss in tightness in the proofs. As observed in Section 4.2, the scheme supports an estimated number of users around $3^{(2^{K_m-1}-1)}/(n/2)$. In practice, for $K_m = 6$ the number of supported users is 2^{40} for $n = 2^{10}$ and 2^{39} for $n = 2^{11}$. The set $\bar{\mathcal{U}}$ results to have cardinality 2^{1805} (resp. 2^{1829}) for $n = 2^{11}$ (resp. $n = 2^{12}$). Hence to compensate for complexity leveraging we consider SVP instances that offer $\sim 2^{1850}$ bits of security (as in the proof of Theorem 7 both $\bar{\mathbf{m}}$ and $\bar{\mathbf{c}}$ have to be guessed).

In the following, we give an example of how we computed the storage requirements in Table 7.3. Consider the parameters in the first line of Table 7.3 (classical security, no complexity leveraging). With those values, a polynomial $\mathbf{a} \in \mathcal{R}_q$ can be stored in at most $n \log_2 q/8 = 11.264$ KB. The issuer public key contains by $\mathbf{a} \in \mathcal{R}_q$ and $\mathbf{B}, \mathbf{C} \in \mathcal{R}_q^{1\times m}$. Hence it is composed by 27 polynomials in $\mathcal{R}_q$, and it requires 304.128 KB of storage. The issuer secret key is composed by the trapdoor $\mathbf{X} \in \mathcal{R}_q^{2\times m}$ sampled from a Gaussian with standard deviation $\sigma_t = 4$, thus its components have infinity norm less than $8 \cdot 4 = 32$ with high probability. Therefore, storing it requires at most $2mn \cdot \log_2 32/8 = 16.64$ KB. The user secret key is composed by the identity $\mathbf{m}$ and the signature $\mathbf{S}$, thus it can be stored in 98.24 KB. A signature is composed by a commitment, i.e. a vector in $\mathcal{R}_q^{1\times m}$, and by the transcript of the NIZK proof (a challenge in $\mathcal{C}$ and two responses, i.e. vectors in $\mathcal{R}_q$ of length $m+2$ and $2m+3$ respectively). The length of the commitment is $mn \log_2 q$ bits that is 146.432 KB. The length of each proof is at most 0.225 MB plus the one-time signature. Hence the length of the presentation token is less than 1.5 MB plus the one-time signature used in the relaxed Σ-protocol.

7.4 Simple Optimization.

We discuss a simple optimization that allows to reduce the size of the token. We left it out when presenting the scheme for ease of exposition, as it would have made harder to grasp the core ideas without adding new insights on the construction.

When using complexity leverage, parameters get considerably larger because the set $\bar{\mathcal{U}}$ can contain up to 2^{1538} elements. Its dimension is determined by the norm of the vector $\mathbf{Z}_1$ in Section 6. This length in turn is a function of the length of $\mathbf{m}$, $\mathbf{E}$ and $\mathbf{b}$, where the norms of $\mathbf{b}$ and $\mathbf{E}$ are considerably larger than the norm of $\mathbf{m}$, as $\mathbf{m} \in \mathcal{R}_3^{(2^{K_m})}$. Hence, it can happen that the norm of an extracted message $\bar{\mathbf{m}}$ is heavily dependent on the norm of $\mathbf{b}$ and $\mathbf{E}$. To avoid this, we can modify the relaxed protocol in Section 6 to do rejection sampling separately on $\mathbf{m}$. Indeed, the prover can sample the error polynomial $\mathbf{y}_m$ from a Gaussian with standard variation σ_m proportional to the norm of $\mathbf{m}$, and another masking vector $\mathbf{Y}_1$ from a Gaussian with standard deviation σ proportional to the norm of $[\mathbf{b}; \mathbf{E}]$. Then, one would do everything as in the original algorithm except for the rejection sampling part. This part would have to be done separately for $\mathbf{z}_m = \mathbf{y}_m + \mathbf{cm}$ and $\mathbf{Z}_1 = \mathbf{Y}_1 + \mathbf{c}[\mathbf{b}; \mathbf{E}]$. The advantage is that now we have a tighter bound on the norm of $\mathbf{z}_m$. The disadvantage is that we have to do three rejection sampling steps (one on $[\mathbf{S}_1; \mathbf{S}_2; \mathbf{S}_3; -\mathbf{E}\mathbf{S}_3]$, one on $[\mathbf{b}; \mathbf{E}]$ and one on $\mathbf{m}$) instead of one, thus reducing the acceptance probability. To prevent that, we can increase the standard deviations. In Appendix F we show the optimized protocol in details and present parameters for it. We observed that the improvement is of around 0.3MB for cases in which we do not consider complexity leverage, and 3 MB in case it is considered.

References

[AABN02] M. Abdalla, J. H. An, M. Bellare, C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security, *EUROCRYPT 2002*, 2002.

[ABB10] S. Agrawal, D. Boneh, X. Boyen. Efficient lattice (H)IBE in the standard model, *EUROCRYPT 2010*, 2010.

[ACJT00] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme, *CRYPTO 2000*, 2000.

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. Post-quantum key exchange: A New Hope., *USENIX Security Symposium 2016*, 2016.

[AFG+10] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-preserving signatures and commitments to group elements, *CRYPTO 2010*, 2010.

[AG11] S. Arora, R. Ge. New algorithms for learning in presence of errors, *ICALP 2011, Part I*, 2011.

[AO09] M. Abe, M. Ohkubo. A framework for universally composable non-committing blind signatures, *ASIACRYPT 2009*, 2009.

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[BBS04] D. Boneh, X. Boyen, H. Shacham. Short group signatures, *CRYPTO 2004*, 2004.

[BCC04] E. F. Brickell, J. Camenisch, L. Chen. Direct anonymous attestation, *ACM CCS 04*, 2004.

[BCJ08] A. Bagherzandi, J. H. Cheon, S. Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma, *ACM CCS 08*, 2008.

[BCK+14] F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures, *ASIACRYPT 2014, Part I*, 2014.

[BCKL08] M. Belenkiy, M. Chase, M. Kohlweiss, A. Lysyanskaya. P-signatures and noninteractive anonymous credentials, *TCC 2008*, 2008.

[BHJ+15] F. Böhl, D. Hofheinz, T. Jager, J. Koch, C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. Classical hardness of learning with errors, *45th ACM STOC*, 2013.

[BMW03] M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions, *EUROCRYPT 2003*, 2003.

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

[CDNZ11] J. Camenisch, M. Dubovitskaya, G. Neven, G. M. Zaverucha. Oblivious transfer with hidden access control policies, *PKC 2011*, 2011.

[CFN90] D. Chaum, A. Fiat, M. Naor. Untraceable electronic cash, *CRYPTO'88*, 1990.

[CHK+11] J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, V. Naessens. Structure preserving CCA secure encryption and applications, *ASIACRYPT 2011*, 2011.

[CKL+16] J. Camenisch, S. Krenn, A. Lehmann, G. L. Mikkelsen, G. Neven, M. Ø. Pedersen. Formal treatment of privacy-enhancing credential systems, *SAC 2015*, 2016.

[CKY09] J. Camenisch, A. Kiayias, M. Yung. On the portability of generalized Schnorr proofs, *EUROCRYPT 2009*, 2009.

[CL01] J. Camenisch, A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation, *EUROCRYPT 2001*, 2001.

[CL03] J. Camenisch, A. Lysyanskaya. A signature scheme with efficient protocols, *SCN 02*, 2003.

[CN11] Y. Chen, P. Q. Nguyen. BKZ 2.0: Better lattice security estimates, *ASIACRYPT 2011*, 2011.

[CNR12] J. Camenisch, G. Neven, M. Rückert. Fully anonymous attribute tokens from lattices, *SCN 12*, 2012.

[CNs07] J. Camenisch, G. Neven, a. shelat. Simulatable adaptive oblivious transfer, *EUROCRYPT 2007*, 2007.

[Coh13] H. Cohen. *A course in computational algebraic number theory*. Springer Science & Business Media, 2013.

[CS03] J. Camenisch, V. Shoup. Practical verifiable encryption and decryption of discrete logarithms, *CRYPTO 2003*, 2003.

[Dam02] I. Damgård. On σ-protocols. *Lecture Notes, University of Aarhus, Department for Computer Science*, 2002.

[DF02] I. Damgård, E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order, *ASIACRYPT 2002*, 2002.

[DM14] L. Ducas, D. Micciancio. Improved short lattice signatures in the standard model, *CRYPTO 2014, Part I*, 2014.

[Fis06] M. Fischlin. Round-optimal composable blind signatures in the common reference string model, *CRYPTO 2006*, 2006.

[FKMV12] S. Faust, M. Kohlweiss, G. A. Marson, D. Venturi. On the non-malleability of the Fiat-Shamir transform, *INDOCRYPT 2012*, 2012.

[FS87] A. Fiat, A. Shamir. How to prove yourself: Practical solutions to identification and signature problems, *CRYPTO'86*, 1987.

[GMW86] O. Goldreich, S. Micali, A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract), *27th FOCS*, 1986.

[GN08] N. Gama, P. Q. Nguyen. Predicting lattice reduction, *EUROCRYPT 2008*, 2008.

[GPV08] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions, *40th ACM STOC*, 2008.

[GS08] J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups, *EUROCRYPT 2008*, 2008.

[HPS11] G. Hanrot, X. Pujol, D. Stehlé. Terminating BKZ. Cryptology ePrint Archive, Report 2011/198, 2011. http://eprint.iacr.org/2011/198.

[HS00] M. Hirt, K. Sako. Efficient receipt-free voting based on homomorphic encryption, *EUROCRYPT 2000*, 2000.

[KF15] P. Kirchner, P.-A. Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices, *CRYPTO 2015, Part I*, 2015.

[KY05] A. Kiayias, M. Yung. Group signatures with efficient concurrent join, *EUROCRYPT 2005*, 2005.

[Lam79] L. Lamport. Constructing digital signatures from a one-way function. Technical report, Technical Report CSL-98, SRI International Palo Alto, 1979.

[LLLS13] F. Laguillaumie, A. Langlois, B. Libert, D. Stehlé. Lattice-based group signatures with logarithmic signature size, *ASIACRYPT 2013, Part II*, 2013.

[LLM+16] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions, *ASIACRYPT 2016*, 2016.

[LLNW14] A. Langlois, S. Ling, K. Nguyen, H. Wang. Lattice-based group signature scheme with verifier-local revocation, *PKC 2014*, 2014.

[LLNW16] B. Libert, S. Ling, K. Nguyen, H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors, *EUROCRYPT 2016, Part II*, 2016.

[LM06] V. Lyubashevsky, D. Micciancio. Generalized compact knapsacks are collision resistant, *Automata, Languages and Programming*. Springer, 2006.

[LN17] V. Lyubashevsky, G. Neven. One-shot verifiable encryption from lattices. 2017.

[LNW15] S. Ling, K. Nguyen, H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based, *PKC 2015*, 2015.

[LPR13] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43, 2013. Preliminary version appeared in EUROCRYPT 2010.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures, *ASIACRYPT 2009*, 2009.

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

[Lyu16] V. Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings, *ASIACRYPT 2016, Part II*, 2016.

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

[NZZ15] P. Q. Nguyen, J. Zhang, Z. Zhang. Simpler efficient group signatures from lattices, *PKC 2015*, 2015.

[PR06] C. Peikert, A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices, *TCC 2006*, 2006.

[PZ] C. Paquin, G. Zaverucha. U-prove cryptographic specification v1. 1 (revision 3, december 2013).

[Ste94] J. Stern. A new identification scheme based on syndrome decoding, *CRYPTO'93*, 1994.

[XLL08] R. Xue, N. Li, J. Li. Algebraic construction for zero-knowledge sets. *J. Comput. Sci. Technol.*, 23(2):166–175, 2008.

Acknowledgements

Working on this paper, we have enjoyed many discussions with Vadim Lyubashevsky. Thank you! This work was supported by the ERC under grant #321310 PERCY) and the SNF under grant #200021_157080 (Efficient Lattice-Based Cryptographic Protocols).

A Standard Definitions

A.1 Sigma Protocols

We recall the standard correctness and zero-knowledge properties of Σ-protocols.

Definition 7. *A Σ-protocol $\Sigma = ((\mathcal{P}_0, \mathcal{P}_1), (\mathcal{V}_0, \mathcal{V}_1))$ for relation R is said to satisfy correctness and honest-verifier zero-knowledge (HVZK) if the following properties hold:*

Correctness. *For all $(x, w) \in R$, if $\mathcal{P}(x, w)$ and $\mathcal{V}(x)$ follow the protocol, the verifier always accepts:*

$$\Pr\left[b = 1 \ : \ \begin{array}{c} \alpha \leftarrow \mathcal{P}_0(x, w; \rho) \ ; \ \beta \leftarrow \mathcal{V}_0(x) \ ; \\ \gamma \leftarrow \mathcal{P}_1(x, w, \alpha, \beta; \rho) \ ; \ b \leftarrow \mathcal{V}_1(x, \alpha, \beta, \gamma) \end{array}\right] \ = \ 1 \ ,$$

where the probability is over the coins ρ of $\mathcal{P}_0$ and the coins of $\mathcal{V}_0$.

Honest-verifier zero knowledge (HVZK). *There exists an efficient algorithm S, called zero-knowledge simulator, such that for any PPT distinguisher D and for any $(x, w) \in R$,*

$$\Pr\left[b' = b \ : \ \begin{array}{c} b' \leftarrow \{0,1\} \ ; \ \alpha \leftarrow \mathcal{P}_0(x, w; \rho) \ ; \\ \beta \leftarrow \mathcal{V}_0(x) \ ; \ \gamma \leftarrow \mathcal{P}_1(x, w, \alpha, \beta; \rho) \ ; \\ \pi_0 \leftarrow (\alpha, \beta, \gamma) \ ; \ \pi_1 \leftarrow \mathsf{S}(x) \ ; \ b' \leftarrow \mathsf{D}(x, w, \pi_b) \end{array}\right] - \frac{1}{2}$$

is negligible.

The Fiat-Shamir transformation of an interactive protocol $(\mathcal{P}, \mathcal{V})$ is a non-interactive zero-knowledge (NIZK) proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$ using a random oracle $\mathcal{H}_c$ with range equal to the space of the verifier's coins, where the proving algorithm $\mathcal{P}^{\mathcal{H}_c}(x, w)$ computes a proof $\pi = (\alpha, \beta, \gamma)$ by choosing random coins ρ and computing $\alpha \leftarrow \mathcal{P}_0(x, w; \rho)$, $\beta \leftarrow \mathcal{H}_c(x, \alpha)$, and $\gamma \leftarrow \mathcal{P}_1(x, w, \alpha, \beta; \rho)$. The verification algorithm $\mathcal{V}^{\mathcal{H}_c}(x, \pi)$ checks that $\mathcal{H}_c(x, \alpha) = \beta$ and $\mathcal{V}_1(x, \alpha, \beta, \gamma) = 1$.

A zero-knowledge simulator S of a (relaxed) NIZK proof system is a stateful algorithm that can operate in two modes: $(h_i, st) \leftarrow \mathsf{S}(1, st, q)$ answers random oracle queries $\mathcal{H}_c(q)$, while $(\pi, st) \leftarrow \mathsf{S}(2, st, x)$ simulates a NIZK proof π for x. Below, the oracle $\mathsf{S}_1(q)$ returns the first outputs of $\mathsf{S}(1, st, q)$, the oracle $\mathsf{S}_2(x, w)$ returns the first output of $\mathsf{S}(2, st, x)$ if $(x, w) \in R$ and returns $\perp$ otherwise, and oracle $\mathsf{S}'_2(x)$ returns the first output of $\mathsf{S}(2, st, x)$ regardless whether $x \in L$ or not.

Definition 8 (Relaxed NIZK). *A* relaxed NIZK proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$ *in the random-oracle model for relations* $(R, \bar{R})$ *is couple of PPT algorithms with the following properties:*

Correctness. *For all* $(x, w) \in R$ *it holds that* $\mathcal{V}^{\mathcal{H}_c}(x, \mathcal{P}^{\mathcal{H}_c}(x, w)) = 1$ *with probability one.*

Unbounded non-interactive zero-knowledge. *There exists a PPT simulator* S *such that for all PPT distinguishers* D *the following quantity is negligible:*

$$\left|\Pr\left[\mathsf{D}^{\mathcal{H}_c(\cdot), \mathcal{P}^{\mathcal{H}_c}(\cdot,\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathsf{D}^{\mathsf{S}_1(\cdot), \mathsf{S}'_2(\cdot,\cdot)}(1^\lambda) = 1\right]\right| \ . \tag{13}$$

Relaxed unbounded simulation soundness. *There exists a PPT simulator* S *such that for all PPT adversaries* A*,*

$$\Pr\left[\mathcal{V}^{\mathsf{S}_1}(x^*, \pi^*) = 1 \wedge x^* \notin \bar{L} \wedge (x^*, \pi^*) \notin Q \ : \ (x^*, \pi^*) \leftarrow \mathsf{A}^{\mathsf{S}_1, \mathsf{S}'_2}(1^\lambda)\right]$$

is negligible, where Q *is the set of tuples* (x, π) *where* A *made a query* $\mathsf{S}_2(x)$ *and obtained response* π*.*

A.2 Commitments

A commitment scheme must satisfy the standard correctness property, i.e., for all $M \in \mathcal{U}$

$$\Pr\left[\mathsf{OpenVf}(cpar, c, M, o) = 1 \ : \ \begin{array}{l} cpar \leftarrow \mathsf{ComParGen}(\mathcal{U}, 1^\lambda), \\ (c, o) \leftarrow \mathsf{Commit}(cpar, M) \end{array}\right] = 1 \ .$$

The hiding property ensures that a commitment value does not reveal information about the committed message.

Definition 9 (Hiding). *A relaxed commitment scheme* rC *is hiding if for all polynomial-time* A

$$\left|\Pr\left[b' = b \ : \ \begin{array}{l} cpar \leftarrow \mathsf{ComParGen}(\mathcal{U}, 1^\lambda), \ (M_0, M_1, st) \leftarrow \mathsf{A}(cpar), \\ b \leftarrow \{0, 1\}, \ (c, o) \leftarrow \mathsf{Commit}(cpar, M_b), \ b' \leftarrow \mathsf{A}(st, c) \end{array}\right] - \frac{1}{2}\right|$$

is negligible.

A.3 Anonymous Attribute Tokens without Opening

Here we recall the definition of anonymous attribute tokens [CNR12], which can be seen as simplified anonymous credentials. They allow users to obtain a credential from an issuer containing a list of attributes. They can then selectively disclose subsets of these attributes to verifiers in such a way that not even the verifier and the issuer together can link different presentations by the same user.

In this section, we focus on AAT schemes without opening (AAT-O), i.e., without a trusted opener who can de-anonymize presentation tokens. In Appendix G, we also provide a construction of an AAT scheme with opening (AAT+O) using verifiable encryption [LN17], which immediately gives rise to a group signature scheme.

A.4 Definition of AAT-O Schemes

We assume that each user can obtain only one credential from each issuer. That credential would then contain all the attributes that the issuer will ever issue to that user. Alternatively, one can see the user identity as a credential identity that binds together the attributes of that credential, but hand multiple such credentials to the same user.

System parameters generation. The public parameters of the scheme are generated from the security parameter as $par \leftarrow \mathsf{SPGen}(1^\lambda)$. They are common to all the parties.

Issuer Key Generation. An issuer generates a public key ipk and corresponding secret key isk by running $\mathsf{IKGen}(par)$.

Credential issuance. To issue a credential for attributes $(\alpha_i)_{i=1}^{\ell}$ to a user, the issuer samples a user identity id, checks that id is not in the list $\mathcal{S}$ of issued user identities (otherwise, he aborts) and runs $\mathsf{Issue}(isk, id, (\alpha_i)_{i=1}^{\ell})$. He hands the resulting user identity id and credential $cred$ to the user.

Presentation. A user creates a presentation token pt revealing a subset of attributes $(\alpha_i)_{i\in R}$, $R \subseteq \{1, \ldots, \ell\}$, from a credential while authenticating a message M by running $\mathsf{Present}(ipk, cred, R, M)$.

Verification. The verifier checks the validity of a presentation token by running $\mathsf{Verify}(ipk, R, (\alpha_i)_{i\in R}, M, pt)$ which returns *accept* or *reject*.

Correctness requires that if the above algorithms are executed honestly, then Verify returns *accept* with probability one for all user identities id, all $\ell \in \mathbb{N}$, all $\alpha_1, \ldots, \alpha_\ell \in \{0,1\}^*$, and all sets $R \subseteq \{1, \ldots, \ell\}$.

Unforgeability. The advantage of an adversary A in breaking the unforgeability of an AAT-O scheme is defined as the probability that it wins the following game. The experiment runs SPGen to generate the public parameters par, IKGen to generate the issuer's secret and public keys isk^* and ipk^*. It then runs A on input par, ipk^* and gives it access to the following oracles:

Issuance to corrupt user: On input attributes $(\alpha_i)_{i=1}^{\ell}$, the experiment samples a user identity id and checks if $id \notin \mathcal{S}$. If this is the case, it runs $\mathsf{Issue}(isk^*, id, (\alpha_i)_{i=1}^{\ell})$, returns id with the resulting credential $cred$. Finally, id is added to $\mathcal{S}$, the set of the assigned user identities id.

Issuance to honest user: On input attributes $(\alpha_i)_{i=1}^{\ell}$, the experiment samples a user identity id, checks that $id \notin \mathcal{S}$ and runs $cred \leftarrow \mathsf{Issue}(isk^*, id, (\alpha_i)_{i=1}^{\ell})$, stores $id, cred$ with a unique credential identifier cid, adds id to $\mathcal{S}$, and returns cid to A.

Presentation by honest user: On input credential identifier cid, opener public key opk, revealed attribute set $R \subseteq \{1, \ldots, \ell\}$ and message M, the experiment checks that a credential $cred$ with identifier cid exists. If so, then it returns $pt \leftarrow \mathsf{Present}(ipk^*, opk, cred, R, M)$ to A.

Eventually, A outputs a presentation token pt^* together with $R^*, (\alpha_i^*)_{i \in R^*}, M^*$. The adversary wins if $\mathsf{Verify}(ipk^*, R^*, (\alpha_i^*)_{i \in R^*}, M^*, pt^*) = 1$ and no credential for attributes $(\alpha_i^*)_{i \in R^*}$ was ever issued to a corrupt user identity id, and no presentation by an honest user was ever performed for attributes $(\alpha_i^*)_{i \in R^*}$ and message M^*.

Anonymity. The anonymity experiment generates $par \leftarrow \mathsf{SignParGen}(1^\lambda)$. The adversary is run on input par and is given access to the following oracle:

Challenge: This oracle can only be queried once with input $ipk^*, cred_0^*, cred_1^*, R^*, (\alpha_i^*)_{i \in R^*}, M^*$. It generates two presentation tokens $pt_0^* \leftarrow \mathsf{Present}(ipk^*, cred_0^*, R^*, M^*)$ and $pt_1^* \leftarrow \mathsf{Present}(ipk^*, cred_1^*, R^*, M^*)$ and checks that both tokens are valid, i.e., $\mathsf{Verify}(ipk^*, R^*, (\alpha_i^*)_{i \in R^*}, M^*, pt_0^*) = \mathsf{Verify}(ipk^*, R^*, (\alpha_i^*)_{i \in R^*}, M^*, pt_1^*) = accept$. If so, then it chooses a random bit $b \leftarrow \{0,1\}$ and returns pt_b^* to A.

The adversary wins the game if it outputs $b' = b$.

B Relaxed NIZKs from One-Time Signatures

A *One-Time Signature (OTS) scheme* for message set $\mathcal{M}$ is composed by a key generation algorithm $(sk, vk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$, a signing algorithm $\sigma \leftarrow \mathsf{OTS.Sign}(sk, msg)$ and a verification algorithm $0, 1 \leftarrow \mathsf{OTS.Vf}(vk, msg, \sigma)$.

Correctness requires that for all security parameters $\lambda \in \mathbb{N}$ it holds that:

$$\mathsf{Pr}\big[1 \leftarrow \mathsf{OTS.Vf}\,(vk, msg, \sigma) \;\mid\; (sk, vk) \leftarrow \mathsf{OTS.Gen}(1^\lambda),\ \sigma \leftarrow \mathsf{OTS.Sign}(sk, msg)\big] = 1\ .$$

A OTS scheme is said to be strongly unforgeable under chosen-message attacks if a PPT adversary has negligible probability in winning the following game:

Experiment $\mathsf{Exp}^{\mathsf{sufcma}}_{\mathsf{A}}(\lambda)$
$(sk, vk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$msg^* \leftarrow \mathsf{A}(\lambda, vk)$
$\sigma^* \leftarrow \mathcal{O}_{OTS}(msg^*)$
$(msg', \sigma') \leftarrow \mathsf{A}(\lambda, vk, (msg^*, \sigma^*))$
If $1 \leftarrow \mathsf{OTS.Vf}\,(vk, msg', \sigma')$
and $(msg', \sigma') \neq (msg^*, \sigma^*)$,
then return 1 else return 0.

Oracle $\mathcal{O}_S(msg)$
$\sigma \leftarrow \mathsf{OTS.Sign}(sk, msg)$
Return σ .

In particular, a OTS that is still secure even against a quantum computer and that can be used with our relaxed Σ-protocol is the Lamport signature [Lam79].

In the following we give the full proof that a relaxed Σ-protocol with the Fiat-Shamir transformation and a one-time signature results in a relaxed NIZK proof. We first rephrase the non-triviality definition for identification schemes due to Abdalla et al. [AABN02] in terms of Σ-protocols.

Definition 10 (Non-trivial min-entropy of commitment). *Let λ be the security parameter, let $\mathsf{Coins}(\lambda)$ the set of coins used by the prover, and let $A(x, w) = \{\mathcal{P}_0(x, w; \rho) \ : \ \rho \leftarrow \mathsf{Coins}(\lambda)\}$ for any $(x, w) \in R$. The min-entropy of $(\mathcal{P}, \mathcal{V})$ is defined as $\varepsilon(\lambda) = \min_{(x,w)\in R} \log_2(1/\mu(x, w))$, where $\mu(x, w)$ is the maximum probability that the first round of the protocol takes on a particular value, i.e. $\mu(x, w) = \max_{\alpha\in\mathsf{A}(x,w)} \mathsf{Pr}[\mathcal{P}_0(x, w; \rho) = \alpha \ : \ \rho \leftarrow \mathsf{Coins}(\lambda)]$. We say that $(\mathcal{P}, \mathcal{V})$ is* non-trivial *if $\varepsilon(n) = \omega(\log(\lambda))$.*

The Σ-protocol described in Section 3 is non-trivial since for a standard deviation $\sigma \geq \omega(\sqrt{\log(\lambda)})$, it holds that $\mathcal{D}_{\mathbb{Z}^{nm},\sigma} \leq \frac{1+\epsilon}{1-\epsilon} 2^{-\lambda}$, hence the min-entropy function is greater than n (cfr. [PR06], Lemma 2.11).

Theorem 13. *Let $(\mathcal{P}, \mathcal{V})$ be a relaxed Σ-protocol for a NP language L. Let $\mathcal{H}_c$ be a hash function with range equal to the space of the verifier's coins, modeled as a random oracle and let $(\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Vf})$ be a strongly unforgeable one-time signature scheme. Then the proof system $(\mathcal{P}^{\mathcal{H}_c}, \mathcal{V}^{\mathcal{H}_c})$, derived from $(\mathcal{P}, \mathcal{V})$ as described in Section 3, is unbounded non-interactive zero-knowledge.*

Proof. Correctness is straightforward. The proof of NIZK property is a modification of the proof of Theorem 1 in Faust et al. [FKMV12], and holds also for classical Σ-protocols. The simulator works as follows:

- To answer a query q to S_1, $\mathsf{S}(1, st, q)$ samples a lookup table $\mathcal{T}_H$ kept in state st and returns $\mathcal{T}_H(q)$. If $\mathcal{T}_H(q)$ is not defined, it sets it to a fresh random value (of the appropriate length).
- To answer a query x to S'_2, $\mathsf{S}(2, st, x)$ calls the HVZK simulator of $(\mathcal{P}, \mathcal{V})$ to obtain the transcript (α, β, γ). Then, it generates the keys of the OTS $(sk, vk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$. Finally, it computes $\sigma \leftarrow \mathsf{OTS.Sign}(sk, (\alpha, \beta, \gamma))$ and it updates $\mathcal{T}_H$ so that $\beta = \mathcal{T}_H(x, \alpha, vk)$. If $\mathcal{T}_H$ was already defined on this input it returns $\perp$ and aborts.

By construction, the only case in which a distinguisher D can distinguish with non-negligible probability the simulation from the real-world protocol is when the simulator S'_2 aborts. This can happen only if (x, α, vk) was already queried to S_1. Given that the protocol is non-trivial, the probability of an abort is upper-bounded by $2^{-\varepsilon(\lambda)}$, that is negligible in λ. Hence:

$$\begin{aligned}
&\mathsf{Pr}\left[\mathsf{D}^{\mathsf{S}_1(\cdot),\mathsf{S}'_2(\cdot,\cdot)}(1^\lambda) = 1\right] = \\
&= \mathsf{Pr}\left[\mathsf{D}^{\mathsf{S}_1(\cdot),\mathsf{S}'_2(\cdot,\cdot)}(1^\lambda) = 1 \mid \mathsf{S}'_2 \text{ aborts}\right] \mathsf{Pr}\left(\mathsf{S}'_2 \text{ aborts}\right) + \\
&\quad + \mathsf{Pr}\left[\mathsf{D}^{\mathsf{S}_1(\cdot),\mathsf{S}'_2(\cdot,\cdot)}(1^\lambda) = 1 \mid \mathsf{S}'_2 \text{ does not abort}\right] \mathsf{Pr}\left(\mathsf{S}'_2 \text{ does not abort}\right) \\
&= 0 + \Pr\left[\mathsf{D}^{\mathcal{H}_c(\cdot),\mathcal{P}^{\mathcal{H}_c}(\cdot,\cdot)}(1^\lambda) = 1\right] \mathsf{Pr}(\mathsf{S}'_2 \text{ does not abort}) \\
&\leq \Pr\left[\mathsf{D}^{\mathcal{H}_c(\cdot),\mathcal{P}^{\mathcal{H}_c}(\cdot,\cdot)}(1^\lambda) = 1\right] (1 - 2^{-\varepsilon(n)}) \ .
\end{aligned}$$

Therefore the difference in (13) is bounded by $2^{-\varepsilon(\lambda)}$ that is negligible in λ.

Finally, we prove relaxed unbounded simulation soundness. Let A be an adversary that breaks the unbounded relaxed simulation soundness property with probability ϵ. Let S be the simulator previously described. At the end of the querying phase, the adversary outputs a valid forgery (x^*, π^*) that was not in the set Q of pairs queried to the simulator. Let the forged proof be $\pi^* = ((\alpha^*, vk^*), \beta^*, (\gamma^*, \sigma^*))$. Without loss of generality, we can assume that A queried (x^*, α^*, vk) to S_1.

Now, we can have two cases: either A learned α^* and vk^* by querying x^* to the simulator S'_2, or it did not. We indicate by query the case in which A queried for x^* and $\overline{\mathsf{query}}$ the other case. Given that these two events are mutually exclusive, the probability that A wins the forgery game can be split in:

$$\Pr\left[\mathsf{A} \text{ wins}\right] = \Pr\left[\mathsf{A} \text{ wins } \wedge \ \mathsf{query}\right] + \Pr\left[\mathsf{A} \text{ wins } \wedge \ \overline{\mathsf{query}}\right] \ .$$

We treat each case separately. In particular, we build two reductions $\mathsf{B}_{\mathsf{s-unf}}$ and $\mathsf{B}_{\mathsf{sound}}$ that exploit A to break either the strong-unforgeability of the OTS or the relaxed soundness of the relaxed Σ-protocol.

Assume that A wins and query happens. The reduction $\mathsf{B}_{\mathsf{s-unf}}$ has access to an oracle $\mathcal{O}_{OTS}$ that on input a message outputs a pair of keys and a signature on the message. To break strong-unforgeability of the OTS scheme, $\mathsf{B}_{\mathsf{s-unf}}$ should output a different signature valid with respect to the same public key. To do that, it first guesses the query j' to S'_2 for which A will forge. Then, it works as follows:

Queries to S_1. It answers to queries to S_1 and fills the table $\mathcal{T}_H$ as the real simulator would have done.

Queries to S'_2. Upon receiving the j-th query:

- If $j \neq j'$, $\mathsf{B}_{\mathsf{s-unf}}$ answers and stores the pair query-answer in the list Q as the real simulator S would do.
- If $j = j'$, $\mathsf{B}_{\mathsf{s-unf}}$ generates the transcript $(\alpha', \beta', \gamma')$ as the real HVZK simulator of the relaxed Σ-protocol $(\mathcal{P}, \mathcal{V})$ would do. It triggers $\mathcal{O}_{OTS}$ to receive the key pair (sk', vk'). Then, it queries $\mathcal{O}_{OTS}$ with $(x*, \alpha', \beta', \gamma')$

to obtain a signature σ'. $\mathsf{B}_{\mathsf{s-unf}}$ sets $\mathcal{T}_H(x^*, \alpha', vk') = \beta'$ and updates Q accordingly. If the value was already assigned, $\mathsf{B}_{\mathsf{s-unf}}$ aborts and outputs $\perp$. Otherwise, it stores $(x^*, (\alpha', vk', \beta', \gamma', \sigma'))$ in *Queries* and outputs $\pi' = ((\alpha', vk'), \beta', (\gamma', \sigma'))$.

At the end of the querying phase, A outputs a valid forgery (x^*, π^*). Remark that, for it to be a valid forgery, it should hold $\pi' \neq \pi^*$.

The reduction $\mathsf{B}_{\mathsf{s-unf}}$ parses the response as (γ^*, σ^*), where σ^* is a signature on the message $msg^* = (x^*, \alpha^*, \beta^*, \gamma^*)$. Then it recovers from Q the query $(x^*, (\alpha', vk', \beta', \gamma', \sigma'))$. Let $msg' = (x^*, \alpha', \beta', \gamma')$. Given that query happened, it holds that $\alpha' = \alpha^*$ and $vk^8 = vk'$, and subsequently $\beta' = \beta^*$. Hence, we can have two cases:

- If $\gamma' = \gamma^*$ (i.e., if $msg' = msg^*$), then for π^* to be a valid forgery it should be $\sigma^* \neq \sigma'$. Hence (msg^*, σ) is a valid forgery.
- If $\gamma' \neq \gamma^*$ then σ^* is a signature on a different message with respect to the key vk. Hence (msg^*, σ^*) is again a valid forgery.

Finally, observe that as before the probability that $\mathsf{B}_{\mathsf{s-unf}}$ aborts when simulating S_2' is negligible as the scheme is non-trivial. Hence, $\mathsf{B}_{\mathsf{s-unf}}$ outputs (msg^*, σ^*) (and breaks strong-unforgeability) with probability $\mathsf{Pr}[\mathsf{A} \text{ wins } \wedge \mathsf{query}] \cdot (1 - 2^{-\varepsilon(\lambda)})\frac{1}{n_q}$, where n_q is the number of queries that A can ask to the simulator. Therefore it holds $\Pr[\mathsf{A} \text{ wins } \wedge \mathsf{query}] \sim \nu(\lambda)$.

Now, consider the case in which A wins and it did compute α^* or vk^* by itself. We build $\mathsf{B}_{\mathsf{sound}}$ as follows. It first chooses an index $j' \in \{1, \ldots, n_q\}$ uniformly at random and then it implements the simulator as follows:

Queries to S_1. Upon receiving query (x_j, α_j, vk_j):
- If $j \neq j'$, $\mathsf{B}_{\mathsf{sound}}$ answers the query and fills the table $\mathcal{T}_H$ as the real simulator would have done.
- If $j = j'$, $\mathsf{B}_{\mathsf{sound}}$ runs the relaxed Σ-protocol with the honest verifier $\mathcal{V}$ for statement $x_{j'}$ with commitment $\alpha_{j'}, vk_{j'}$. Upon receiving the challenge β from $\mathcal{V}$, it programs $\mathcal{T}_H(x_j, \alpha_j, vk_{j'}) = \beta$. Then it outputs β as an answer to the query.

$\mathsf{B}_{\mathsf{sound}}$ answers as the real simulator would to the queries of type $(x, \mathfrak{a})$.

Queries to S_2'. To answer these queries, the reduction honestly executes the NIZK simulator. Upon receiving query x, $\mathsf{B}_{\mathsf{sound}}$ runs $\mathsf{OTS.Gen}(1^\lambda)$ to obtain a key pair (sk, vk). Then, it runs $\mathsf{OTS.Sign}(sk, (x, \alpha, \beta, \gamma))$ to obtain a signature σ. $\mathsf{B}_{\mathsf{sound}}$ sets $\mathcal{T}_H(x, \alpha, vk) = \beta$ and updates Q accordingly. If the value was already assigned, $\mathsf{B}_{\mathsf{sound}}$ aborts and outputs $\perp$. Otherwise, it stores $(x, (\alpha, vk, \beta, \gamma, \sigma))$ in Q and outputs $\pi = ((\alpha, vk), \beta, (\gamma, \sigma))$.

Finally, when A outputs the forgery $(x^*, (\alpha^*, vk^*, \beta^*, \gamma^*, \sigma^*))$ the simulator parses and sends γ^* to the $\mathcal{V}$. The success probability is bounded by:

$$\mathsf{Pr}[\mathsf{B}_{\mathsf{sound}} \text{ wins}] = \Pr[\mathsf{A} \text{ wins } \wedge \overline{\mathsf{query}}] \frac{1}{n_q} \ .$$

Thus,

$$\Pr[\mathsf{A} \text{ wins } \wedge \overline{\mathsf{query}}] = \mathsf{Pr}[\mathsf{B}_{\mathsf{sound}} \text{ wins}] n_q \leq \nu(\lambda)$$

and hence we have that:

$$\epsilon = \Pr[\mathsf{A} \text{ wins}] = \Pr[\mathsf{A} \text{ wins } \wedge \ \mathsf{query}] + \Pr[\mathsf{A} \text{ wins } \wedge \ \overline{\mathsf{query}}] \leq \nu(\lambda) \ .$$

□

C Generalized Forking Lemma

Let A be an adversary, let IG be an input generator for A, let $f = (\rho, h_1, \ldots, h_{q_\mathsf{H}})$ be A's random coins and random-oracle responses, and let $f_j = (\rho, h_1, \ldots, h_j)$. An execution of A is successful if it returns a non-empty set of indices $J \subseteq \{1, \ldots, q_\mathsf{H}\}$ and corresponding outputs $\{sig_j\}_{j \in J}$. Let Ω be the set of all f and let Ω_{in} be the set of f for which A is successful on input in; its success probability is $\varepsilon = \Pr[f \in \Omega_{in} : in \leftarrow \mathsf{IG}, f \xleftarrow{\$} \Omega]$. Consider the following generalized forking algorithm:

Algorithm $\mathsf{GF}_\mathsf{A}(in)$:
$f \xleftarrow{\$} \Omega$
$(J, \{sig_j\}_{j \in J}) \leftarrow \mathsf{A}(in; f)$
If $J = \emptyset$ then halt
Let $J = \{j_1, \ldots, j_n\}$ such that $j_1 \leq \ldots \leq j_n$, $X \leftarrow \{(h_j, sig_j)\}_{j \in J}$, $X' \leftarrow \emptyset$
For $i = 1, \ldots, n$ do
  $succ_i \leftarrow 0$, $k_i \leftarrow 0$, $k_{\max} \leftarrow 8n\,q_\mathsf{H}/\varepsilon \cdot \ln(8n/\varepsilon)$
  Repeat until $succ_i = 1$ or $k_i > k_{\max}$
    $k_i \leftarrow k_i + 1$
    $f' = (\rho', h'_1, \ldots, h'_{q_\mathsf{H}}) \leftarrow \{f' \in \Omega : f'_j = f_j\}$
    $(J', \{sig'_j\}_{j \in J'}) \leftarrow \mathsf{A}(in; f')$
    If $J' \neq \emptyset \ \wedge \ j_i \in J' \ \wedge \ h'_{j_i} \neq h_{j_i}$ then $X' \leftarrow X' \cup \{(h'_{j_i}, sig'_{j_i})\}$, $succ_i \leftarrow 1$
If $succ_1 = \ldots = succ_n = 1$ then return (X, X') else return $\perp$

Lemma 7 (Generalized forking lemma [BCJ08]). *If algorithm* A *runs in time* t *and has success probability* ε*, then the forking algorithm* GF_A *runs in time* $t \cdot 8n^2 q_\mathsf{H}/\varepsilon \cdot \ln(8n/\varepsilon)$ *and returns* (X, X') *with probability* $\hat{\varepsilon} \geq \varepsilon/8$*.*

D A Useful Lemma

Lemma 8. *Let* $\mathbf{a}, \ \mathbf{b} \in \mathcal{R}_q$ *be such that* $n\|\mathbf{a}\|_\infty \cdot \|\mathbf{b}\|_\infty \leq (q-1)/2$*. Then we have*

$$\|\mathbf{ab}\| \leq \|\mathbf{a}\|\|\mathbf{b}\|\sqrt{n} \ \wedge \ \|\mathbf{ab}\|_\infty \leq \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty n \leq \frac{q-1}{2} \ .$$

Proof. Let $\mathbf{a} = \sum_{i=0}^{n-1} a_i \mathbf{x}^i$ and $\mathbf{b} = \sum_{i=0}^{n-1} b_i \mathbf{x}^i$. The product $\mathbf{ab}$ can be represented as a matrix-polynomial product $A \cdot B$, where $B \in \mathbb{Z}_q^n$ is a column vector$[b_0; b_1; \ldots; b_{n-1}]$ and $A \in \mathbb{Z}_q^{n \times n}$ is a matrix whose columns are the column vectors corresponding to the polynomials $\mathbf{a}$, $\mathbf{xa}$, $\ldots$ $\mathbf{x}^{n-1}\mathbf{a}$:

$$A \cdot B = \begin{bmatrix} a_0 & -a_{n-1} & \ldots & -a_1 \\ a_1 & a_0 & \ldots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \ldots & a_0 \end{bmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix} \bmod q \ .$$

Let A_j be the j-th row of A. The coefficient of $\mathbf{x}^j$ in the product $\mathbf{ab}$ is the scalar product $\langle A_j, B\rangle$. Moreover, observe that $\|\mathbf{a}\| = \|A_1\| = \|A_j\|$ for $j = 2, \ldots, n$, as the Euclidean norm is sign-invariant (the same holds for the infinity norm). From these observations and from the Cauchy-Schwarz inequality it follows that:

$$\|\mathbf{ab}\| = \sqrt{\sum_{j=1}^{n} \langle A_j, B\rangle^2} \leq \sqrt{\sum_{j=1}^{n} \|A_j\|^2 \|B\|^2} = \|\mathbf{a}\|\|\mathbf{b}\|\sqrt{n} \ ,$$

where the Cauchy-Schwarz inequality holds as the hypothesis on the infinity norms of $\mathbf{a}$ and $\mathbf{b}$ guarantees that there is no rounding $\bmod q$ in the computation of the coefficients of $\mathbf{ab}$. The second bound follows from the observation that $\|AB\|_\infty = \max_j \langle A_j, B\rangle \leq \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty n$. □

E Indistinguishability Result for the Signature Scheme

In this Appendix we prove that an algorithm A against Assumption 3 cannot distinguish B from a challenger behaving as specified by Assumption 3. The result is stated in the following.

Theorem 14. *Assume there exists an adversary* A *is able to distinguish the simulator* B *in the proof of Theorem 7 from a honest signer. Then there exists a distinguisher* D *that can distinguish* $(\mathbf{A}, \mathbf{U})$ *for uniformly sampled* $\mathbf{U}$ *from* $(\mathbf{A}, \mathbf{AR})$ *for* $\mathbf{R} \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q, s}$ *and* s *is either* σ_t *or* σ *with the same probability and running time of* A.

Observe that $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$, hence indistinguishability relies on Ring-LWE, that is a computational assumption. It would be also possible to have statistical indistinguishability by adapting the results from [GPV08] to rings, but this would require $\mathbf{A}$ to have a larger dimension (namely, $\mathbf{A}$ should be in $\mathcal{R}_q^{1\times \ell}$ where $\ell \geq 2\log(q) \sim 200$).

The hardness of the underlying Ring-LWE instance can be estimated computing the Hermite Root factors as it follows. The secret is a vector sampled from $\mathcal{D}^m_{\mathcal{R}_q, s}$, hence by Lemma 3 its norm is bounded by $B = 1.05s\sqrt{nm}$. We can approximate the Hermite Root factor as $\delta = B^{1/(nm)}/q^{1/(nm^2)}$. With the parameter choices in Table 7.3 this value is smaller than the Hermite root factor of the Ring-SIS instance in Theorem 7 for both σ_t and σ.

To make the proof easier to understand, we split the result in three lemmas. The first ensures that, upon receiving the public parameters from the simulator B, A cannot distinguish them from honestly generated ones.

Lemma 9. *Let* n *be a power of* 2, q *a prime such that* $q \equiv 5 \bmod 8$ *and* $\sigma_t > 0$. *Assume that there exists a polynomial-time algorithm* A *able to distinguish between the following two distributions:*

- ***distribution 1:*** $\mathbf{A}$ *uniformly sampled matrix in* $\mathcal{R}_q^{1\times 2}$, $\mathbf{B} = \mathbf{AR} + \mathbf{G} \in \mathcal{R}_q^{1\times m}$ *where* $\mathbf{R} \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q, \sigma_t}$, $\mathbf{C}$ *uniformly sampled in* $\mathcal{R}_q^{1\times m}$.

– ***distribution 2:*** $\mathbf{A}$ *uniformly sampled matrix in* $\mathcal{R}_q^{1\times 2}$, $\mathbf{B}$ *uniformly sampled in* $\mathcal{R}_q^{1\times m}$, $\mathbf{C} = \mathbf{AR} - \mathbf{mc}^{-1}\mathbf{G}$ *where* $\mathbf{R} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$, $\mathbf{m}$ *is sampled uniformly at random in* $\mathcal{R}_{2p+1}^{(2^{K_m})}$, $\mathbf{G}$ *is the gadget matrix and* $\mathbf{c} \xleftarrow{\$} \bar{\mathcal{C}}$, *where* p *is defined as in Section 5.2.*

Then there exists a distinguisher D *that is able to distinguish the distributions* $(\mathbf{A}, \mathbf{U})$ *for* $\mathbf{U} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *and* $(\mathbf{A}, \mathbf{AR})$ *for* $\mathbf{R} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$ *exploiting* A *with the same success probability.*

Proof. We prove the result with a sequence of indistinguishable game hops.

Game 0. In Game 0 everything is generated as in distribution 1.

Game 1. In Game 1 everything is generated as in Game 0 except for $\mathbf{B}$, that it is now chosen uniformly at random in $\mathcal{R}_q^{1\times m}$. Distinguishing Game 1 from Game 0 is exactly equivalent to distinguish the distributions $(\mathbf{A}, \mathbf{U})$ and $(\mathbf{A}, \mathbf{AR})$ where $\mathbf{R}$ is a matrix sampled from $\mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$.

Game 2. In Game 2 everything is generated as in distribution 2. Let A be an algorithm able to distinguish Game 2 from Game 1. We construct a reduction D that can distinguish between $(\mathbf{A}, \mathbf{U})$ and $(\mathbf{A}, \mathbf{AR})$ for $\mathbf{R} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$ exploiting A with the same success probability.

Let $(\mathbf{A}, \mathbf{U})$ be the instance of the decisional problem D has to solve. D samples $\mathbf{m} \xleftarrow{\$} \mathcal{R}_{2p+1}^{(2^{K_m})}$ and $\mathbf{c} \xleftarrow{\$} \bar{\mathcal{C}}$. By Lemma 1 $\mathbf{c}$ is invertible , thus D can set $\mathbf{C} = \mathbf{U} - \mathbf{mc}^{-1}\mathbf{G}$. Then it sends $(\mathbf{A}, \mathbf{C})$ to A and it outputs *distribution 1* if A returns *uniform*, and *distribution 2* otherwise. If $\mathbf{U}$ was sampled uniformly in $\mathcal{R}_q^{1\times m}$ then also the components of $\mathbf{U} - \mathbf{mc}^{-1}\mathbf{G}$ are distributed uniformly in $\mathcal{R}_q$. Indeed,

$$
\begin{aligned}
&\mathsf{Pr}(\mathbf{U} - \mathbf{mc}^{-1}\mathbf{G} = \mathbf{K}) = \\
&= \sum_{\mathbf{c}',\mathbf{m}'} \mathsf{Pr}(\mathbf{U} - \mathbf{mc}^{-1}\mathbf{G} = \mathbf{K} \,|\mathbf{c} = \mathbf{c}',\ \mathbf{m} = \mathbf{m}')\mathsf{Pr}(\mathbf{c} = \mathbf{c}',\ \mathbf{m} = \mathbf{m}') \\
&= \sum_{\mathbf{c},\mathbf{m}'} \mathsf{Pr}(\mathbf{U} = \mathbf{K} + \mathbf{m}'\mathbf{c}'^{-1}\mathbf{G})\frac{1}{|\bar{\mathcal{C}}|}\frac{1}{|\mathcal{R}_{2p+1}^{(2^{K_m})}|} \;=\; \frac{1}{|\mathcal{R}_q^{1\times m}|}
\end{aligned}
$$

where the second equation holds because $\mathbf{c}$ and $\mathbf{m}$ are independent and $\mathbf{U}$ is independent from $\mathbf{m}$ and $\mathbf{c}$. Otherwise, $\mathbf{U} = \mathbf{AR}$ and the distribution is exactly distribution 2.

□

Now, we prove that, even after the querying phase, A cannot distinguish D from the challenger. We again split the result in two lemmas, depending on whether A queried $(\mathbf{m}, \alpha)$ where $\mathbf{m} = \bar{\mathbf{c}_1}^{-1}\bar{\mathbf{m}}$.

Lemma 10 ($\mathbf{m} = \bar{\mathbf{c}_1}^{-1}\bar{\mathbf{m}}$). *Let* n *be a power of* 2, q *a prime such that* $q \equiv 5 \bmod 8$ $\sigma_t > 0$ *and* σ *as defined in Section 5.2. Let* $\mathbf{G}$ *be the gadget vector,*

$\mathcal{H} : \{0,1\}^* \longmapsto \mathcal{R}_q$ *be a random element of the family of hash functions defined on* $\{0,1\}$ *with values in* $\mathcal{R}_q$*. Let* $\mathbf{A} = [\mathbf{a}|1]$ *for* $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$*. Assume there exists a polynomial-time algorithm* A *able to distinguish between the two following distributions:*

- ***distribution 1****: vector* $[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{U}]$ *where* $\mathbf{U} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$ *and* $\mathbf{R} \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q,\sigma_t}$*, access to an oracle* $\mathcal{O}_H$ *that on input* $\alpha \in \{0,1\}^*$ *outputs* $\mathcal{H}(\alpha)$*, and to an oracle* $\mathcal{O}_S$ *where* $\mathcal{O}_S(\mathbf{m},\alpha)$ *outputs an* $\mathbf{S} \sim \mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{U}+\mathbf{mG}],\mathcal{H}(\alpha),\sigma}$*.*
- ***distribution 2****: vector* $[\mathbf{A}|\mathbf{B}|\mathbf{AR}']$ *where* $\mathbf{B}$ *is a random vector in* $\mathcal{R}_q^{1\times m}$ *and* $\mathbf{R}' \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q,\sigma_t}$*, access to an oracle* $\mathcal{O}_H{}'$ *that on input* $\alpha \in \{0,1\}^*$ *outputs* $[\mathbf{A}|\mathbf{B}|\mathbf{AR}']\mathbf{S}$ *for a* $\mathbf{S}$ *sampled from* $\mathcal{D}^{\perp}_{\mathcal{R}_q^{2+2m},\mathbf{0}_{2+2m},\sigma}$*, and to an oracle* $\mathcal{O}_S{}'$ *where* $\mathcal{O}_S{}'(\alpha)$ *outputs the* $\mathbf{S}$ *s.t.* $\mathcal{H}(\alpha) = [\mathbf{A}|\mathbf{B}|\mathbf{AR}']\mathbf{S}$*.*

Then there exists another algorithm D *that is able to distinguish the distribution* $(\mathbf{A},\mathbf{AR})$ *from* $(\mathbf{A},\mathbf{U})$*, where* $\mathbf{U}$ *is uniformly sampled from* $\mathcal{R}_q^{1\times m}$ *and* $\mathbf{R}$ *comes from a Gaussian with standard deviation in* $\{\sigma_t,\sigma\}$*.*

Proof. We prove the claim with a sequence of games.

Game 0. In Game 0 everything is constructed as in distribution 1.

Game 1. In Game 1, everything is constructed as in distribution 1 except for the vector, that is constructed as $[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{B}+\mathbf{G}]$. Game 1 is statistically indistinguishable from Game 0 as if $\mathbf{B}$ is uniformly distributed, also $\mathbf{B}+\mathbf{G}$ is.

Game 2. In Game 2, everything is constructed as in Game 1 except for the vector, that is constructed as $[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{AR}'+\mathbf{G}]$ for some $\mathbf{R}' \xleftarrow{\$} \mathcal{D}^{2\times m}_{\mathcal{R}_q,\sigma_t}$. Game 2 is computationally indistinguishable from Game 1 if the distribution of $(\mathbf{A},\mathbf{AR})$ is indistinguishable from $(\mathbf{A},\mathbf{U})$. Remark that Lemma 5 guarantees that we can sample from $\mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{AR}'+\mathbf{G}],\mathcal{H}(\alpha),\sigma}$ using both $\mathbf{R}$ and $\mathbf{R}'$ as trapdoor.

Game 3. In Game 3, everything is constructed as in Game 2 except for the vector, that is constructed as $[\mathbf{A}|\mathbf{B}+\mathbf{G}|\mathbf{AR}'+\mathbf{G}]$ for some $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$. Game 3 is indistinguishable from Game 2 if $(\mathbf{A},\mathbf{AR})$ is indistinguishable from $(\mathbf{A},\mathbf{U})$. We can still sample from $\mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{B}+\mathbf{G}|\mathbf{AR}'+\mathbf{G}],\mathcal{H}(\alpha),\sigma}$ using $\mathbf{R}'$ as trapdoor.

Game 4. In Game 4, everything is constructed as in Game 3 except for the vector, that is constructed as $[\mathbf{A}|\mathbf{B}|\mathbf{AR}'+\mathbf{G}]$ for some $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$. Game 5 is statistically indistinguishable from Game 4, as $\mathbf{B}$ and $\mathbf{B}+\mathbf{G}$ are both distributed as a uniform if $\mathbf{B}$ is. We can still sample from $\mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{B}|\mathbf{AR}'+\mathbf{G}],\mathcal{H}(\alpha),\sigma}$ using $\mathbf{R}'$ as trapdoor.

Game 5. In Game 5, everything is constructed as in Game 4 except for the hash. On input α, a vector $\mathbf{S} \xleftarrow{\$} \mathcal{D}^{2+2m}_{q,\sigma}$ is sampled and the output of $\mathcal{O}_H(\alpha)$ is $[\mathbf{A}|\mathbf{B}|\mathbf{AR}']\mathbf{S}$. Signing is still done using the trapdoor hidden in the public key. Again, if an adversary could distinguish this from a uniformly sampled vector,

then a simulator can exploit it to distinguish $(\mathbf{A}, \mathbf{AR})$ from $(\mathbf{A}, \mathbf{U})$, where in this case $\mathbf{R}$ comes from a Gaussian with standard deviation σ. Thus Game 5 is computationally indistinguishable from Game 4.

Game 6. In Game 6, the vector is constructed as $[\mathbf{A}|\mathbf{B}|\mathbf{AR}]$ and the oracle $\mathcal{O}_S$ is programmed so that for each input α outputs the $\mathbf{S} \xleftarrow{\$} \mathcal{D}_{q,\sigma}^{2+2m}$ sampled to calculate $\mathcal{H}(\alpha) = [\mathbf{A}|\mathbf{B}|\mathbf{AR}]\mathbf{S}$. The vector $[\mathbf{A}|\mathbf{B}|\mathbf{AR}]$ in Game 6 is indistinguishable from the vector in Game 5 if $(\mathbf{A}, \mathbf{AR})$ is indistinguishable from $(\mathbf{A}, \mathbf{U})$. The distribution of the outputs of $\mathcal{O}_S$ is indistinguishable from the distribution of $\mathbf{S}$ sampled from $\mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{B}|\mathbf{AR}'+\mathbf{G}],\mathcal{H}(\alpha),\sigma}$. Indeed Lemma 5.2 in [GPV08] adapted to rings guarantees that in both cases the distribution of the output $\mathbf{S}$ is $D_{\mathcal{R}_q^{2+2m},\mathbf{0}_{2+2m},\sigma}$.

□

Lemma 11 ($\mathbf{m} \neq \bar{\mathbf{c}_1}^{-1}\bar{\mathbf{m}}$). *Let n be a power of 2, q a prime such that $q \equiv 5$ mod 8, $\sigma_t > 0$ and σ as defined in Section 5.2. Assume for some pair $(\mathbf{m}^*, \mathbf{c}^*) \in \mathcal{R}_{2p+1}^{(2^{K_m})} \times \bar{\mathcal{C}}$, $\mathbf{A} = [\mathbf{a}|\mathbf{1}]$ where $\mathbf{a}$ is uniformly random in $\mathcal{R}_q^{1\times 2}$, and hash function $\mathcal{H} : \{0,1\}^* \to \mathcal{R}_q$ there exists a polynomial-time algorithm A which can distinguish the following two distributions:*

- ***distribution 1:*** *vector $[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{U}]$ where $\mathbf{R} \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$, $\mathbf{U} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$, access to an oracle $\mathcal{O}_H$ that on input $\alpha \in \{0,1\}^*$ outputs $\mathcal{H}(\alpha)$, and to an oracle $\mathcal{O}_S$ that on input $(\mathbf{m}, \alpha)$ outputs $\mathbf{S} \sim \mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{U}+\mathbf{mG}],\mathcal{H}(\alpha),\sigma}$ for all $\mathbf{m} \in \mathcal{U}$ such that $\mathbf{m} - \mathbf{m}^*\mathbf{c}^{*-1}$ is invertible and $\sigma \approx q^{1/m} \cdot s_1(\mathbf{R})$.*
- ***distribution 2:*** *vector $[\mathbf{A}|\mathbf{B}|\mathbf{AR}' - \mathbf{m}^*\mathbf{c}^{*-1}\mathbf{G}]$ where $\mathbf{R}' \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_q,\sigma_t}^{2\times m}$, $\mathbf{B}$ is uniformly random in $\mathcal{R}_q^{1\times m}$, access to an oracle $\mathcal{O}_H{}'$ that on input $\alpha \in \{0,1\}^*$ outputs $[\mathbf{A}|\mathbf{B}|\mathbf{AR}']\mathbf{S}$ for a $\mathbf{S}$ sampled from $\mathcal{D}^{\perp}_{\mathcal{R}_q^{2+2m},\mathbf{0}_{2+2m},\sigma}$, and to an oracle $\mathcal{O}_S$ where $\mathcal{O}_S(\mathbf{m}, \alpha)$ queries α to the previous oracle to obtain $\mathbf{u} = \mathcal{O}_H(\alpha)$ and outputs an $\mathbf{S} \sim \mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{U}|\mathbf{AR}'-\mathbf{m}^*\mathbf{c}^{*-1}\mathbf{G}+\mathbf{mG}],\mathbf{u},\sigma}$ for all $\mathbf{m} \in \mathcal{U}$ such that $\mathbf{m} - \mathbf{m}^*\mathbf{c}^{*-1}$ is invertible and $\sigma \approx q^{1/m} \cdot s_1(\mathbf{R})$.*

Then there exists another algorithm D which is able to distinguish the distribution $(\mathbf{A}, \mathbf{AR})$ from $(\mathbf{A}, \mathbf{U})$, where $\mathbf{U}$ is uniformly sampled from $\mathcal{R}_q^{1\times m}$ and $\mathbf{R}$ comes from a Gaussian with standard deviation in $\{\sigma_t, \sigma\}$.

Proof. The proof is a sequence of indistinguishable games.

Game 0. The starting Game is distribution 1, where $\mathcal{O}$ is implemented by using the "trapdoor" $\mathbf{R}$ as in Theorem 1.

Game 1. The second game replaces the $\mathbf{U}$ by $\mathbf{U} - \mathbf{m}^*\mathbf{c}^{*-1}\mathbf{G}$. Since $\mathbf{U}$ is uniformly random, so is $\mathbf{U} - \mathbf{m}^*\mathbf{c}^{*-1}\mathbf{G}$, and thus Games 1 and 2 are identical.

Game 2. Game 2 replaces the $\mathbf{U}$ with $\mathbf{AR}'$ (and so we have the vector $[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{AR}' - \mathbf{m}^*\mathbf{c}^{*-1}\mathbf{G}]$), where $\mathbf{R}'$ has the same distribution as $\mathbf{R}$. If there is a polynomial-time algorithm that can distinguish Game 2 from Game 1, then one can distinguish between distributions $(\mathbf{A}, \mathbf{U})$ and $(\mathbf{A}, \mathbf{AR}')$.

Game 3. Game 3 is exactly like Game 2 except $\mathcal{O}$ is now implemented by using $\mathbf{R}'$ as the "trapdoor". Since $\mathbf{m} \neq \mathbf{m}^*\mathbf{c}^{*-1}$ one can again use Theorem 1 to generate an element from the distribution $\mathcal{D}^{\perp}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}|\mathbf{AR}'-(\mathbf{m}-\mathbf{m}^*\mathbf{c}^{*-1})\mathbf{G}],\mathcal{H}\alpha,\sigma}$ using $\mathbf{R}'$ as a trapdoor as long as $\mathbf{m} - \mathbf{m}^*\mathbf{c}^{*-1}$ is invertible. By Lemma 5 we know that using $\mathbf{R}$ or $\mathbf{R}'$ as a trapdoor produces the same distribution. Thus the distributions of the outputs of Games 2 and 3 are the same.

Game 4. Game 4 replaces $\mathbf{AR} + \mathbf{G}$ in the public key with $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{1\times m}$. If there is algorithm that can distinguish between Game 3 and 4, then it can distinguish between distributions $(\mathbf{A}, \mathbf{U})$ and $(\mathbf{A}, \mathbf{AR})$.

Game 5. Finally, in Game 5 everything is constructed as in Game 4 except for the hash oracle. On input α, a vector $\mathbf{S} \xleftarrow{\$} \mathcal{D}_{q,\sigma}^{2+2m}$ is sampled and the output of $\mathcal{O}_H(\alpha)$ is $[\mathbf{A}|\mathbf{B}|\mathbf{AR}']\mathbf{S}$. Again, if an adversary could distinguish this from a uniformly sampled vector, then a simulator can exploit it to distinguish $(\mathbf{A}, \mathbf{AR})$ from $(\mathbf{A}, \mathbf{U})$ where now $\mathbf{R}$ comes from a Gaussian with standard deviation σ. Thus Game 5 is computationally indistinguishable from Game 4. Observe that Game 5 is exactly *distribution 2*.

□

F Relaxed Proofs of Signatures on Partially Hidden Messages with Multiple Rejection Sampling Steps

In this section we discuss how to generate the presentation token performing rejection sampling separately on signature, opening information and message $\mathbf{m}$. We only consider the case in which the prover has a single signature-message pair. The general case can be obtained following the same procedure presented at the end of Section 6.

Let $\mathbf{S}$ be a signature on a message $(\mathbf{m}, \alpha)$, and $\mathbf{F}$ a commitment to $\mathbf{m}$ with opening information $[\mathbf{b}; \mathbf{E}]$. As in Section 6, the prover can rearrange the terms of the verification equation of the signature to prove knowledge of both $\mathbf{S}$ and $\mathbf{m}$. In particular, the prover constructs the vectors $\mathbf{S}_c$ and $\mathbf{S}_s$ as:

$$\text{(I)}\ \underbrace{\left[-\mathbf{G}^T\ \mathbf{F}^T\ -\mathbb{I}_m\right]}_{=\mathbf{A}_c}\underbrace{\begin{pmatrix}\mathbf{m}\\ \mathbf{b}\\ \mathbf{E}^T\end{pmatrix}}_{=\binom{\mathbf{m}}{\mathbf{S}_c}} = \mathbf{C}^T\ ,\ \text{(II)}\ \underbrace{[\mathbf{A}|\mathbf{B}|\mathbf{F}|\mathbf{1}]}_{=\mathbf{A}_s}\underbrace{\begin{pmatrix}\mathbf{S}_1\\ \mathbf{S}_2\\ \mathbf{S}_3\\ -\mathbf{E}\mathbf{S}_3\end{pmatrix}}_{=\mathbf{S}_s} = \mathcal{H}(\alpha)\ . \quad (14)$$

Observe that now $\mathbf{S}_c$ does not include $\mathbf{m}$. To define the relations for the relaxed Σ-protocol, we bound the norm of the secrets as it follows. The norm of the message $\mathbf{m}$ is bounded by $N_1 = \sqrt{2^{2K_m-1}}$, as it is an element of $\mathcal{R}_3^{(2^{K_m})}$ of degree $n/2$. The vector $\mathbf{S}_c$ is composed by $1+m$ polynomials in $\mathcal{R}_3$: its norm is bounded by $N_2 = \sqrt{(1+m)n}$. For the norm of $\mathbf{S}_s$, the first $2+2m$ components have norm bounded by $1.05\sigma\sqrt{n}$, while the norm of the last component can be

bound using the properties of the singular value:

$$\|\mathbf{E}\mathbf{S}_3\| \leq \|\mathbf{E}\| \cdot \mathsf{s}_1(\mathbf{S}_3) \leq \sqrt{nm} \cdot \frac{\sigma}{\sqrt{\pi}}\sqrt{n}(1+\sqrt{m}+\log n) \ .$$

Set $N_3 = \sqrt{(2+2m)(1.05\sigma\sqrt{n})^2 + (n\frac{\sigma\sqrt{m}}{\sqrt{\pi}}(1+\sqrt{m}+\log n))^2}$ and define the following relations (for fixed constants $\bar{N}_1$, $\bar{N}_{1,\infty}$, $\bar{N}_2$, $\bar{N}_{2,\infty}$):

$$\begin{aligned} R = \{((\mathbf{A}_c, \mathbf{C}^T, \mathbf{A}_s, \mathcal{H}(\alpha)), (\mathbf{m}, \mathbf{S}_c, \mathbf{S}_s, \mathbf{1})) \ : \ & \mathbf{m} \in \mathcal{U} \qquad (15) \\ & (\mathbf{m}, \mathbf{S}_c, \mathbf{1}) \text{ satisfy relation (I) and } \|\mathbf{S}_c\| \leq N_1 \ , \ \|\mathbf{S}_c\|_\infty \leq 1 \\ & (\mathbf{S}_s, \mathbf{1}) \text{ satisfy relation (II) and } \|\mathbf{S}_s\| \leq N_2 \ , \ \|\mathbf{S}_s\|_\infty < (q-1)/(2n)\} \\ \bar{R} = \{((\mathbf{A}_c, \mathbf{C}^T, \mathbf{A}_s, \mathcal{H}(\alpha)), (\bar{\mathbf{m}}, \bar{\mathbf{S}}_c, \bar{\mathbf{S}}_s, \bar{\mathbf{c}})) \ : \ & \bar{\mathbf{m}} \in \bar{\mathcal{U}} \ , \ \|\bar{\mathbf{m}}\| < \bar{N}_1 \qquad (16) \\ & (\bar{\mathbf{m}}, \bar{\mathbf{S}}_c, \bar{\mathbf{c}}) \text{ satisfy relation (I) and : } \|\bar{\mathbf{S}}_c\| \leq \bar{N}_1 \ , \ \|\bar{\mathbf{S}}_c\|_\infty \leq \bar{N}_{1,\infty} \\ & (\bar{\mathbf{S}}_s, \bar{\mathbf{c}}) \text{ satisfy relation (II) and : } \|\bar{\mathbf{S}}_s\| \leq \bar{N}_2 \ , \ \|\bar{\mathbf{S}}_s\|_\infty \leq \bar{N}_{2,\infty}\} \ . \end{aligned}$$

Let $\mathcal{P}$ and $\mathcal{V}$ be the prover and verifier defined in Section 3.2 w.r.t. the relations (15) and (16). To prove that $\mathbf{m}$ is in $\bar{\mathcal{U}}$ (i.e., $\mathbf{m}$ is an element of $\mathcal{R}_{2p+1}^{(2^{K_m})}$ with degree $3n/4$) set the challenge spaces $\mathcal{C}$ and $\bar{\mathcal{C}}$ to be as in Section 4.2. To have better bounds on the message and on the opening information, $\mathcal{P}$ does rejection sampling separately for $\mathbf{S}_s$, $\mathbf{S}_c$ and $\mathbf{m}$. Hence, $\mathcal{P}$ samples $\mathbf{y}_1 \xleftarrow{\$} \mathcal{D}_{\mathcal{Y}_{3n/m},\sigma_1}$, $\mathbf{Y}_2 \xleftarrow{\$} \mathcal{D}^{1+m}_{\mathcal{R}_q,\sigma_2}$ and $\mathbf{Y}_3 \xleftarrow{\$} \mathcal{D}^{2m+3}_{\mathcal{R}_q,\sigma_2}$ and sends as commitments $(\mathbf{A}_c\binom{\mathbf{y}_1}{\mathbf{Y}_2}, \mathbf{A}_s\mathbf{Y}_3)$. Upon receiving the challenge $\mathbf{c}$ from the verifier, the prover sets $\mathbf{z}_1 = \mathbf{y}_1 + \mathbf{cm}$, $\mathbf{Z}_2 = \mathbf{Y}_2 + \mathbf{c}\mathbf{S}_c$ and $\mathbf{Z}_3 = \mathbf{Y}_3 + \mathbf{c}\mathbf{S}_s$, and does rejection sampling separately on them. If $\sigma_1 = 16T_1$, $\sigma_2 = 24T_2$, $\sigma_3 = 24T_3$ (where T_1, T_2, T_3 are upper bounds on the norm of $\mathbf{cm}$, $\mathbf{c}\mathbf{S}_c$, $\mathbf{c}\mathbf{S}_s$ respectively) the probability that the prover outputs $(\mathbf{z}_1, \mathbf{Z}_2, \mathbf{Z}_3)$ is greater than $1/6$ (cfr. Lemma 4.3, 4.4, 4.5 in [Lyu12] and Section F). To compute the values T_1, T_2 and T_3, we observe that a challenge $\mathbf{c} \in \mathcal{C}$ has norm bound by $\|\mathbf{c}\| \leq \sqrt{2^{2^{K_m-2}-1}}$, hence using Lemma 2 we can set $T_i = N_i\sqrt{n2^{2^{K_m-2}-1}}$ for $i = 1, 2, 3$. To guarantee special soundness, we set $\bar{N}_1 = 2.1\sigma_1\sqrt{n}$, $\bar{N}_2 = 2.1\sigma_2\sqrt{n(1+m)}$, $\bar{N}_3 = 2.1\sigma_3\sqrt{(2+2m)n}$ and $\bar{N}_{1,\infty} = 16\sigma_1$, $\bar{N}_{2,\infty} = 16\sigma_2$, $\bar{N}_{3,\infty} = 16\sigma_3$ as in Section 3.2. Setting $p = \bar{N}_{1,\infty}$ guarantees that $\bar{\mathbf{m}} \in \bar{\mathcal{U}}$.

As in Section 6, in case $K_m = 6$ the proof has to be repeated 6 times to have negligible soundness error.

Theorem 15. *Given N_s as in Section 5.2 and $\bar{N}_s = 8.82(3+2m)\sigma_2\sigma_3 n\sqrt{nm}$, $\bar{N}_{s,\infty} = 512\sigma_2\sigma_3 n$, $\bar{C} = 4.2\sigma_2 n\sqrt{2^{K_m-2}-1}$ and $p = 16\sigma_1$, the protocol $(\mathcal{P}, \mathcal{V})$ is a relaxed Σ-protocol for the following pair of relations:*

$$\begin{aligned} R = \{ \ ((\mathbf{A}_s, \mathcal{H}(\alpha)), (\mathbf{m}, (\mathbf{1}, \mathbf{S}, \mathbf{1}))) \ : \ & \mathbf{m} \in \mathcal{U}, \\ & [\mathbf{A}|\mathbf{B}|\mathbf{1}\mathbf{C} + \mathbf{m}\mathbf{G}|\mathbf{1}]\mathbf{S} = \mathbf{1}\mathcal{H}(\alpha) \text{ and } \|\mathbf{S}\| \leq N_s\} \\ \bar{R} = \{ \ ((\mathbf{A}_s, \mathcal{H}(\alpha)), (\bar{\mathbf{m}}, (\bar{\mathbf{c}}_1, \bar{\mathbf{S}}, \bar{\mathbf{c}}_2))) \ : \ & \bar{\mathbf{m}} \in \bar{\mathcal{U}}, \ \bar{\mathbf{c}}_1 \in \bar{\mathcal{C}}, \ \|\bar{\mathbf{c}}_2\| \leq \bar{C} \\ & [\mathbf{A}|\mathbf{B}|\bar{\mathbf{c}}_1\mathbf{C} + \bar{\mathbf{m}}\mathbf{G}|\mathbf{1}]\bar{\mathbf{S}} = \bar{\mathbf{c}}_2\mathcal{H}(\alpha) \text{ and } \|\bar{\mathbf{S}}\| \leq \bar{N}_s, \ \|\bar{\mathbf{S}}\|_\infty \leq \bar{N}_{s,\infty}\} \end{aligned}$$

under Ring-LWE$_{\mathcal{R}_3}$ with the uniform distribution.

Proof. Correctness is trivial.

			Parameters				Sizes		
Compl. Lev.	Security	δ	n	q	m	σ_t	ipk(KB)	usk(KB)	$token$(MB)
NO	classical	1.003735	2^9	$\sim 2^{90}$	22	4	259.2	75.136	1.09
NO	quantum	1.003488	2^{10}	$\sim 2^{90}$	12	4	288	96.576	1.34
NO	worst-case	1.002926	2^{10}	$\sim 2^{90}$	14	4	334.08	103.36	1.54
YES	classical	1.0005248	2^{11}	$\sim 2^{90}$	43	4	2004.48	411.328	8.91
YES	quantum	1.0004842	2^{11}	$\sim 2^{90}$	47	4	2188.8	421.312	9.71
YES	worst-case	1.0003949	2^{11}	$\sim 2^{90}$	58	4	2695.68	512.424	11.74

Table 2. Table of parameters for the AAT-O with 3 rejection sampling steps.

The HVZK of this protocol can be proved by constructing a simulator very similar to the one in Theorem 2. The simulator is indistinguishable from uniform thanks to the anonymity of the commitment scheme and to rejection sampling.

Finally, we prove special soundness. A knowledge extractor $(\mathsf{E}_1^s, \mathsf{E}_2^s)$ rewinds Π to obtain the vectors $[\bar{\mathbf{b}}; \bar{\mathbf{m}}; \bar{\mathbf{E}}]$, $[\bar{\mathbf{S}}_1; \bar{\mathbf{S}}_2; \bar{\mathbf{S}}_3; \bar{\mathbf{s}}_4]$ and the polynomial $\bar{\mathbf{c}}$ that satisfy equations (I) and (II) in (14). Multiplying equation (II) times $\bar{\mathbf{b}}$ and plugging in $\bar{\mathbf{b}}\mathbf{F} = \bar{\mathbf{c}}\mathbf{C} + \bar{\mathbf{m}}\mathbf{G} + \bar{\mathbf{E}}$ yields:

$$\mathbf{A}(\bar{\mathbf{b}}\bar{\mathbf{S}}_1) + \mathbf{B}(\bar{\mathbf{b}}\bar{\mathbf{S}}_2) + [\bar{\mathbf{c}}\mathbf{C} - \bar{\mathbf{m}}\mathbf{G}](\bar{\mathbf{S}}_3) + (\bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3) = \bar{\mathbf{c}}\bar{\mathbf{b}}\mathcal{H}(\alpha) \qquad (17)$$

The vector $\bar{\mathbf{S}} = [\bar{\mathbf{b}}\bar{\mathbf{S}}_1; \bar{\mathbf{b}}\bar{\mathbf{S}}_2; \bar{\mathbf{S}}_3; \bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3]$ has norm bounded by the norm of $\bar{\mathbf{b}}\bar{\mathbf{s}}_4 - \bar{\mathbf{E}}\bar{\mathbf{S}}_3$. The element with the largest norm is $\bar{\mathbf{E}}\bar{\mathbf{S}}_3$ and applying Lemma 2 we have $\|\bar{\mathbf{E}}\bar{\mathbf{S}}_3\| \leq 2.1\sigma_2\sqrt{n} \cdot 2.1\sigma_3\sqrt{n} \cdot \sqrt{nm}$, where the inequality holds as by Lemma 3 we can bound the product of the infinity norms as $\|\bar{\mathbf{E}}\|_\infty \|\bar{\mathbf{S}}_3\|_\infty n < 16\sigma_2 \cdot 16\sigma_3 n$ that is less than $(q-1)/2$ for our choice of parameters. Hence, setting $\bar{N}_s = (3+2m) \cdot 2(2.1\sigma_2\sqrt{n} \cdot 2.1\sigma_3\sqrt{n} \cdot \sqrt{nm})$ we have that $\|\bar{\mathbf{S}}\| \leq \bar{N}_s$ and, again from Lemma 2 $\|\bar{\mathbf{S}}\|_\infty \leq 2 \cdot 16\sigma_2 \cdot 16\sigma_3 n =: \bar{N}_{s,\infty}$. Observe that $\bar{N}_{s,\infty} > 8\sigma_3 > 8\sigma$, hence the correctness of the signature scheme is guaranteed (see Section 5.2). Moreover, $\bar{\mathbf{c}} \in \bar{\mathcal{C}}$, and, applying again Lemma 2, $\|\bar{\mathbf{c}}\bar{\mathbf{b}}\| \leq \|\bar{\mathbf{c}}\|\|\bar{\mathbf{b}}\|\sqrt{n} \leq 2\sqrt{2^{K_m-2}-1} \cdot 2.1\sigma_2 n =: \bar{C}$. Hence $(\bar{\mathbf{c}}, \bar{\mathbf{S}}, \bar{\mathbf{c}}\bar{\mathbf{b}}) \in \bar{\Sigma}$, i.e. the extractor outputs a valid signature. From point 2 in Lemma 3 the infinity norm of the extracted user's secret key $\bar{\mathbf{m}}$ is less than $16\sigma_1 =: p$. □

The protocol is made non interactive via the construction based on OTS presented in Theorem 13, and can be plugged in straightforwardly in the AAT-O scheme in Section 7.2. Parameters for the resulting AAT-O are shown in Table F. They are computed following the same approach used in Section 7.3.

G Anonymous Attribute Tokens with Opening

We present a construction of an AAT+O scheme by combining our relaxed signature and commmitment scheme with the relaxed verifiable encryption scheme by Lyubashevsky and Neven [LN17]. An AAT+O scheme immediately gives rise to a group signature scheme with chosen-ciphertext anonymity (i.e., where the adversary has access to an opening oracle) by handing each user a credential

of a fixed attribute as a signing key. To sign a message M, the user creates a presentation token for M.

We see our AAT+O scheme mainly as a proof of concept that demonstrates that our framework of relaxed cryptographic primitives, glued together with relaxed Σ-protocols, can be composed generically to build efficient privacy-enhancing protocols. At the same time, we expect that a direct construction that builds on the same principles, but that is optimized for the specific use case, can easily outperform our generic construction. We therefore refrain from suggesting concrete parameter sizes and giving efficiency estimates, but rather leave such numbers to future work.

G.1 Definition of AAT+O Schemes

We slightly adapt the syntax and security notions of AAT+O schemes [CNR12] to a setting where the issuer and opener are separate entities, rather than having a central group manager that performs both roles. The syntax of an AAT+O scheme largely follows that of an AAT-O scheme. The system parameters generation, issuer key generation, and credential issuance are as defined for AAT-O schemes in Appendix A.3. There is an additional opener key generation algorithm $\mathsf{OKGen}(par)$ that generates the opener's key pair (opk, osk). The presentation and verification algorithms take as an additional input the public key opk of the opener who can recover the identity of the user who created the token. Finally, the opening algorithm $id \leftarrow \mathsf{Open}(ipk, osk, R, (\alpha_i)_{i\in R}, M, pt)$ recovers the user's identity.

In terms of security, an AAT+O scheme must satisfy unforgeability, traceability, and anonymity. The unforgeability notion is almost identical to that of AAT-O schemes, except that the adversary includes an opener public key in its honest presentation queries and in its output.

Anonymity. We describe a strong notion of full anonymity, often referred to as CCA2 anonymity, where the adversary is given access to an opening oracle. The experiment generates parameters $par \leftarrow \mathsf{SignParGen}(1^\lambda)$ and opener keys $(opk^*, osk^*) \leftarrow \mathsf{OKGen}(par)$. The adversary is run on input par, opk^* and given access to the following oracles:

Opening: On input $ipk, R, (\alpha_i)_{i\in R}, M, pt$, the oracle returns the user identity $id \leftarrow \mathsf{Open}(ipk, osk^*, R, (\alpha_i)_{i\in R}, M, pt)$.

Challenge: This oracle can only be queried once with input $ipk^*, cred_0^*, cred_1^*, R^*, (\alpha_i^*)_{i\in R^*}, M^*$. It generates two presentation tokens $pt_0^* \leftarrow \mathsf{Present}(ipk^*, opk^*, cred_0^*, R^*, M^*)$ and $pt_1^* \leftarrow \mathsf{Present}(ipk^*, opk^*, cred_1^*, R^*, M^*)$ and checks that $\mathsf{Verify}(ipk^*, opk^*, R^*, (\alpha_i^*)_{i\in R^*}, M^*, pt_0^*) = \mathsf{Verify}(ipk^*, opk^*, R^*, (\alpha_i^*)_{i\in R^*}, M^*, pt_1^*) = accept$. If so, then it chooses a random bit $b \leftarrow \{0,1\}$ and returns pt_b^* to A.

The adversary wins the game if it outputs $b' = b$ and never submitted the target presentation token pt_b^* to the opening oracle.

Traceability. The advantage of adversary A in breaking the traceability of the AAT+O scheme is defined as the probability that it wins the following game. The experiment runs SPGen and IKGen as in the unforgeabilty experiment in Appendix A.3, and also generates the opener's key pair $(opk^*, osk^*) \leftarrow \mathsf{OKGen}(par)$. It then runs A on input par, ipk^*, opk^* and, in addition to the oracles in the unforgeability experiment for AAT-O schemes, gives it access to the following oracle:

Opening: On input a presentation token pt and $ipk, R, (\alpha_i)_{i\in R}, M$, the experiment returns $id \leftarrow \mathsf{Open}(ipk, osk^*, R, (\alpha_i)_{i\in R}, M, pt)$.

At the end of the game, A outputs a presentation token pt^* together with $R^*, (\alpha_i^*)_{i\in R^*}, M^*$. Let $id^* \leftarrow \mathsf{Open}(ipk, osk, R, (\alpha_i)_{i\in R}, M, pt)$. The adversary wins if $\mathsf{Verify}(ipk^*, opk^*, R^*, (\alpha_i^*)_{i\in R^*}, M^*, pt^*) = 1$, no credential for attributes $(\alpha_i^*)_{i\in R^*}$ was ever issued to a corrupt user identity id^*, and no presentation by an honest user with identity id^* was ever performed for attributes $(\alpha_i^*)_{i\in R^*}$ and message M^*.

G.2 Relaxed Verifiable Encryption

A relaxed verifiable encryption scheme for witness relations $R, \bar{R}$ is a tuple of algorithms $(\mathsf{EKg}, \mathsf{Enc}, \mathsf{EVf}, \mathsf{Dec})$ where $\mathsf{EKg}(1^\lambda)$ generates a public and secret key (epk, esk), $\mathsf{Enc}(epk, x, w)$ encrypts a witness w for $(x, w) \in R$ as a ciphertext Γ, $\mathsf{EVf}(epk, x, \Gamma)$ returns 0 or 1 indicating whether Γ encrypts a valid witness for x, and $\mathsf{Dec}(esk, x, \Gamma)$ recovers a witness $\bar{w}$ such that $(x, \bar{w}) \in \bar{R}$.

Soundness requires that if an adversary produces a ciphertext Γ for x such that $\mathsf{EVf}(epk, x, \Gamma) = 1$, then with overwhelming probability, $\mathsf{Dec}(esk, x, \Gamma)$ returns a witness $\bar{w}$ such that $(x, \bar{w}) \in \bar{L}$.

Chosen-ciphertext simulatability requires that there exists a simulator Sim so that an adversary cannot distinguish a real encryption $\mathsf{Enc}(epk, x, w)$ from a simulated ciphertext $\mathsf{Sim}(epk, x)$, even when given access to a decryption oracle. We refer to [LN17] for formal definitions.

The relaxed verifiable encryption scheme in [LN17] is based on Ring-LWE encryption combined with a relaxed Σ-protocol for linear relations over short vectors, i.e. relations of the form

$$R = \{((\mathbf{B}, \mathbf{u}), (\mathbf{m}, \mathbf{1})) \in (R_p^{\ell\times k} \times R_p^\ell) \times (R_p^k \times R_p) \ : \ \mathbf{Bm} = \mathbf{u} \bmod p \ \wedge \mathbf{m} \in S_\gamma^k\}$$

and relaxed language $\bar{L}$ with relation

$$\bar{R} = \{((\mathbf{B}, \mathbf{u}), (\bar{\mathbf{m}}, \bar{\mathbf{c}})) \in (R_p^{\ell\times k} \times R_p^\ell) \times (R_p^k \times R_p) : \\ \mathbf{B}\bar{\mathbf{m}} = \bar{\mathbf{c}}\mathbf{u} \bmod p \ \wedge \ \|\bar{\mathbf{m}}\|_\infty < 6\sigma \ \wedge \ \bar{\mathbf{c}} \in \bar{\mathcal{C}}\} \ ,$$

where $\bar{\mathcal{C}} = \{\mathbf{c} - \mathbf{c}' \ : \ \mathbf{c}, \mathbf{c}' \in \mathcal{C}\}$ for $\mathcal{C} = \{\mathbf{c} \in R : \|\mathbf{c}\|_\infty = 1, \|\mathbf{c}\|_1 \le 36\}$.

The scheme takes as parameters two primes r, q where $r > 2$, and a standard variation σ_e and a set of challenges $\mathcal{C} = \mathcal{R}_5^{(2^{K_c})}$. Decryption is successful as long as

$$(36r + 12)\sigma_e < q/2C_1' \tag{18}$$

where $C_1' = \max_{\mathbf{c}\in\bar{\mathcal{C}}} \|\mathbf{c}\|_1$ (Lemma 3.1 [LN17]). We set the message space to be $\mathcal{R}_3^{(64)}$ as in the AAT-O case. To have negligible soundness error, it is enough to repeat the NIZK proof two times. The relaxed verifiable encryption scheme is sound and chosen-ciphertext simulatable under the Ring-LWE assumption.

G.3 Construction of AAT+O

As described in Section 6, the opening verification equation of the commitment scheme can be rewritten as

$$\underbrace{\begin{bmatrix}-\mathbf{G}^T & \mathbf{F}^T & -\mathbb{I}_m\end{bmatrix}}_{=\mathbf{A}_c} \underbrace{\begin{pmatrix}\mathbf{m}\\ \mathbf{b}\\ \mathbf{E}^T\end{pmatrix}}_{=\mathbf{S}_c} = \mathbf{C}^T \ . \tag{19}$$

One can observe that this is a linear relation that can be used for the verifiable encryption scheme of [LN17]. We can therefore add opening to our AAT-O scheme from Section 7 by including in the presentation token a verifiable encryption of the user identity $\mathbf{m}$ that is committed to in $\mathbf{F}$, so that $\mathbf{m}$ can be recovered by decrypting the ciphertext.

System parameter generation, issuer key generation, and issuance are performed exactly as in the AAT-O scheme from Section 7.2. The other algorithms are described as follows.

Opening Key Generation. The opener generates a key pair $(epk, esk) \leftarrow \mathsf{EKg}(1^\lambda)$ for the verifiable encryption scheme and sets $opk = epk$ and $osk = esk$.

Presentation. To create a presentation token for opener key opk, attributes $(\alpha_i)_{i\in R}$ and message M, the user first proceeds as in the AAT-O scheme, i.e., he creates a commitment $\mathbf{F}$ to $\mathbf{m}$ and generates NIZK proof Π_i that he knows a signatures on the committed message and $i\|\alpha_i$ for $i \in R$ using the protocol from Section 6, whereby he includes the message M in the Fiat-Shamir hash. He then generates a one-time signature key pair $(vk, sk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$ and creates a verifiable ciphertext $\Gamma \leftarrow \mathsf{Enc}(opk, \mathbf{A}_c, \mathbf{S}_c)$ as per Equation (19), while including vk in the Fiat-Shamir hash of the verifiable encryption. Finally, he computes a one-time signature $\sigma \leftarrow \mathsf{OTS.Sign}(sk, (\mathbf{F}, (\Pi_i)_{i\in R}, \Gamma))$ and outputs the presentation token $pt = (\mathbf{F}, (\Pi_i)_{i\in R}, \Gamma, vk, \sigma)$.

Verification. The verifier checks the validity of $(\Pi_i)_{i\in R}$ w.r.t. $\mathbf{F}$ and the message M, checks that $\mathsf{EVf}(opk, \mathbf{A}_c, \Gamma) = 1$ with vk in the Fiat-Shamir hash, and that $\mathsf{OTS.Vf}(vk, (\mathbf{F}, (\Pi_i)_{i\in R}, \Gamma)) = 1$. If all tests pass, he outputs *accept*, otherwise *reject*.

Opening. To open a presentation token pt, the opener first runs the verification algorithm, returning $\perp$ if it rejects. The opener decrypts $(\bar{\mathbf{S}}_c, \bar{\mathbf{c}}) \leftarrow \mathsf{Dec}(osk, \Gamma)$ and recovers $\bar{\mathbf{m}}$ as the first coordinate from $\bar{\mathbf{S}}_c$. He then recovers an irreducible factor $\mathbf{m}$ of degree $n/2$ of $\bar{\mathbf{m}}$ and returns $\mathbf{m}$, or returns $\perp$ if such a factor does not exist. (Note that factoring polynomials in $\mathcal{R}_q$ can be done efficiently.)

G.4 Security

The traceability of the AAT+O scheme follows easily from the unforgeability of the AAT-O scheme in Section 7.2.

Theorem 16 (Traceability). *If the relaxed signature scheme* rS *is unforgeable, the relaxed commitment scheme* rC *is binding, and the Lyubashevsky-Neven relaxed verifiable encryption scheme is sound, then the AAT+O scheme is traceable in the Random Oracle Model.*

Proof (Sketch). Given an adversary A who breaks the traceability of the AAT+O scheme, one can use the generalized forking lemma [BCJ08] (recalled in Section C) on the proofs $(\Pi_i)_{i\in R}$ to extract valid relaxed signatures of $(\bar{\mathbf{m}}, i\|\alpha_i)$ for all revealed attributes, as well as a relaxed opening $(\bar{\mathbf{S}}_{c,i}, \bar{\mathbf{c}}_i)$ for the commitment $\mathbf{F}$. By the soundness of the relaxed verifiable encryption scheme, the decryption of Γ also recovers a possibly different relaxed opening $(\bar{\mathbf{S}'}_c, \bar{\mathbf{c}'})$ for $\mathbf{F}$. Let $\bar{\mathbf{m}}_i$ be the message contained in $\bar{\mathbf{S}}_{c,i}$ and let $\bar{\mathbf{m}'}$ be the message contained in $\bar{\mathbf{S}'}_c$. If for some $i \in R$ the message $\bar{\mathbf{m}}_i$ does not share an $n/2$-degree irreducible factor with $\bar{\mathbf{m}'}$, then one can use A to break the relaxed binding property of the commitment scheme. Otherwise, A can be used to obtain a forgery against the signature scheme using a similar reduction as in the unforgeability proof.

Theorem 17 (Anonymity). *If the relaxed* Σ*-protocol* $\mathsf{r}\Sigma$ *is simulatable, the one-time signature scheme is strongly unforgeable, the Lyubashevsky-Neven verifiable encryption scheme is chosen-ciphertext simulatable, and the relaxed commitment scheme* rC *is hiding, then the AAT+O scheme is anonymous in the Random Oracle Model.*

Proof (Sketch). Consider the following sequence of games:

Game 0: This is the real game, i.e., where A obtains pt_b^* generated with $cred_b^*$.

Game 1: Simulate the proofs Π_i^* in the target presentation token pt^*. This game is indistinguishable from the previous one by the unbounded zero-knowledge property of the relaxed Σ-protocol.

Game 2: Reject all opening queries for presentation tokens $pt = ((\Pi_i)_{i\in R}, \Gamma, vk, \sigma)$ where $\Gamma = \Gamma^*$ included in pt^*. Because vk^* is included in the Fiat-Shamir hash of Γ^*, the fact that $\Gamma = \Gamma^*$ and that Γ is valid according to EVf means that $vk = vk^*$. The only way for the adversary to submit a presentation token $pt \neq pt^*$ that is rejected in this game but not in the previous game is therefore by using $(\mathbf{F}, (\Pi_i)_{i\in R}, \sigma) \neq (\mathbf{F}^*, (\Pi_i^*)_{i\in R}, \sigma^*)$, but any adversary doing so can be used to break the strong one-time unforgeability of the OTS scheme.

Game 3: Use the simulator Sim for the verifiable encryption scheme to generate Γ^* as $\mathsf{Sim}(opk^*, \mathbf{F})$. This game is indistinguishable from the previous one because of the chosen-ciphertext simulatability of the verifiable encryption scheme. Note that in the reduction, we never have to answer opening queries for presentation tokens containing Γ^*, as these were already rejected in Game 2.

Game 4: Generate $\mathbf{F}^*$ as a commitment to a dummy message $\mathbf{m}^* = \mathbf{1}$ instead of $\mathbf{m}_b^*$ in $cred_b^*$. This game hop is indistinguishable due to the hiding property of the relaxed commitment scheme.

In the final game, the bit b isn't used at all in the simulation of A's view, i.e., b is information-theoretically hidden from A.