A Generic Construction of an Anonymous Reputation System and Instantiations from Lattices

Johannes Blömer[1], Jan Bobolz[2], and Laurens Porzenheim[1]

[1] *Paderborn University*
{bloemer,laurens.porzenheim}@upb.de
[2] *University of Edinburgh (work done while at Paderborn University)*
jan.bobolz@ed.ac.uk

Abstract. With an anonymous reputation system one can realize the process of rating sellers anonymously in an online shop. While raters can stay anonymous, sellers still have the guarantee that they can be only be reviewed by raters who bought their product.

We present the first generic construction of a reputation system from basic building blocks, namely digital signatures, encryption schemes, non-interactive zero-knowledge proofs, and linking indistinguishable tags. We then show the security of the reputation system in a strong security model. Among others, we instantiate the generic construction with building blocks based on lattice problems, leading to the first module lattice-based reputation system.

1. Introduction

Reputation systems are crucial for markets to function properly. They are usually a user's only indicator regarding the trustworthiness of a seller, or the quality of a product. Right now, in real-world reputation systems, ratings are centrally controlled (see, for example, Amazon or Yelp ratings) by the reputation system provider (Amazon/Yelp).

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" under the project number 160364472 – SFB 901/3. Additionally, this work was partially funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia.

This means that the reputation system provider has the ability to admit or deny users from the system, censor ratings, inject fake ratings, and trace all raters' identities. Of course, this allows a malicious provider to unilaterally undermine the reputation system, e.g. by censoring inconvenient ratings or by using knowledge of user identities to retaliate against bad ratings.

Cryptographic reputation systems. A *cryptographic* reputation system is a decentralized system in which the roles and abilities of the reputation system provider are either fully replaced by cryptographic mechanisms or at least distributed among multiple parties, with strong anonymity guarantees for users. First, a user registers (once) with the *group manager*, who is tasked with admitting users to the system (essentially to prevent Sybil attacks). Then, when the user buys a product, he receives a *rating token* from an *issuer* (e.g., the seller), certifying that the user is indeed allowed to rate the issuer (to prevent users from rating issuers they have never interacted with). Given the membership certificate from the group manager and the rating token from the issuer, the user can create a *rating signature*. We imagine that the user posts this signatures to a public *reputation board*, enabling other users to view and verify the rating. The rating signature is *anonymous*, meaning that it does not reveal who, of all users who are allowed to rate that issuer, issued this particular rating (preventing retaliation against negatively ratings). However, the *opener* possesses a special key to inspect signatures and reveal the user's identity in case of misuse. Finally, even though rating signatures are otherwise anonymous to the public, anyone can efficiently check whether any two rating signatures have been created by the same user (to prevent the same user from submitting multiple ratings for the same issuer). In this setting, the role of the incentive system provider has been distributed among group manager, issuers, reputation boards. User anonymity is cryptographically guaranteed, but can be revoked by the opener. What we describe here can be seen as (a special case of) the ticket-based approach identified by [GG21].

Desirable construction types. There exists a wealth of constructions of such system in the literature (as surveyed in [GG21]), but they all work in the discrete logarithm setting. With the looming threat of quantum computers, there is a need for constructions that do not rely on the hardness of discrete logarithms and instead rely on some hardness assumption not likely broken by quantum computers, such as lattice-based assumptions. We are aware of only a single lattice-based reputation system in the literature, designed by El Kaafarani, Katsumata, and Solomon [EKKS18].

We can generally distinguish generic constructions from non-generic constructions. A generic construction is a prescription how to plug together (almost) arbitrary instantiations of several basic schemes (e.g., signature schemes, encryption schemes, and non-interactive zero-knowledge proofs (NIZKs)) into a secure reputation system. So far, reputation system constructions have been non-generic, i.e. there is no formally proven way to construct reputation systems from arbitrarily instantiated basic building blocks. Even beyond the lack of an *explicit* generic construction, existing constructions are also quite specific to their (discrete logarithm / lattice) setting. For example, a natural choice for rating tokens would be for the issuer to sign the buying user's public key (thereby giving that user the right to rate). However, in the discrete logarithm setting (e.g., [BJK15, BEJ18]), rating tokens are typically (blind) signatures on the user's *secret key*, instead, because traditionally, it is easier to sign secret keys (which live in $\mathbb{Z}_p$) than public keys (which live in the group $\mathbb{G}$). In the lattice setting, the only known construction [EKKS18] accumulates all buyers' public keys in a Merkle hash tree, which is (relatively) efficient in the lattice setting, but would be absurdly inefficient and borderline impossible to implement in the discrete logarithm setting (considering the need to prove statements in zero-knowledge about the hashes).

1.1. Our contribution

In this paper, we give the first provably secure *generic* construction of a reputation system from digital signatures, public-key encryption, linking indistinguishable tags (LITs), and NIZKs. We formally define security properties and prove that the generic construction (and hence any concrete constructions built from it) fulfills them. Furthermore, we show that this generic construction can be reasonably instantiated in both the lattice setting and the discrete logarithm setting, unifying and drawing parallels between the two settings. In particular, this results in the first reputation system based on *module* lattices. Our construction compares favorably in its privacy properties to the only other lattice-based construction [EKKS18], as discussed later.

Generic construction. The generic construction roughly follows a paradigm similar to the sign-encrypt-prove paradigm [CS97] for group signatures, similar to [BJK15, BEJ18] (but modified to apply to both the lattice and the discrete logarithm setting). The user generates some secret key usk; his public key is $\mathsf{upk} = f(\mathsf{usk})$ for some one-way function f. To join the system, the user obtains a signature ρ on his public key from the group manager. To enable rating an issuer, the user also obtains a signature τ on his public key from the issuer. Given those two signatures, the user composes a rating text rtng and encrypts his public key upk for the opener (who holds the secret key to reveal upk in case of misuse). For technical reasons, the user also encrypts usk under a key that nobody knows the secret key for (a trick comparable to the Naor-Yung paradigm). The user then computes a *linking indistinguishable tag* (LIT) using his secret key usk. The LIT is the gadget that will allow anyone to check whether the user has rated the same issuer twice. We then use the NIZK essentially as a signature of knowledge [CL06] to create a non-interactive proof authenticating the rating text rtng by proving, in zero-knowledge, that the ciphertexts and LIT have been computed correctly, and that his public key has been signed by the group manager and the issuer.

Instantiation in the discrete logarithm setting. In the discrete logarithm setting, we can use LIT tags in the random oracle model of the form $\mathcal{RO}(\mu)^{\mathsf{usk}}$ (note that this is a deterministic tag and hence enables detection of a user rating ipk twice). Because the generic construction signs *public* keys, we use a structure-preserving signature as the signature scheme. Unsurprisingly, encryption can be accomplished with ElGamal and the NIZKs can be instantiated with Schnorr-style protocols together with the Fiat-Shamir heuristic. More details can be found in Section 6.2.

Instantiation with lattices. The instantiation with lattices is more difficult given that the ecosystem for privacy constructions is less mature than in the discrete logarithm setting. We need to instantiate the encryption scheme, the signature scheme, the NIZK, and the linking indistinguishable tag. For more efficiency and flexibility when setting parameters, we generally consider the *module* lattice setting.

For the encryption scheme, the typical choice is between primal and dual Regev encryption (i.e. between putting the LWE error into the public key or into ciphertexts). Primal Regev is more suitable for proving statements about encryptions in zero-knowledge, which is why we choose it for the instantiation. Section 4.2 shows a module version of Regev encryption, optimized for use in zero-knowledge proofs with long plaintexts, putting up with a large public key for the benefit of small-dimension randomness vectors (since public data is cheap, but randomness of part of the proof witness and increases the proof size).

There are several signature schemes based on lattice assumptions. However, we require one that plays nicely with zero-knowledge proofs, for example the signature should not rely on random oracles. Thus, a first idea would be to use the signatures of [LLM^{+}16] or [JRLS22], as they are designed to be compatible with current lattice-based proof systems. However, [LLM^{+}16] present a construction based on unstructured lattices, which is too inefficient compared to a construction from structured lattices. Furthermore, their construction inherently uses a chameleon hash to achieve adaptive security, which increases the complexity of a proof of possession of a signature. On the other hand, the construction of [JRLS22] is a stateful signature that can only sign a limited number of signatures, which does not fit our generic construction. Instead, we look at the stateless signatures of [DM14]. As the other two signatures, it is a tag-based signature scheme and a variant of signatures by [Boy10], but is based on ideal lattices. [DM14] show their signature to be non-adaptively secure and transform it to adaptive security by employing chameleon hashes. We instead show in Section 4.1 that the signature of [DM14] is already adaptively secure by using a proof technique as in [LSS14].

For the NIZK, we chose [LNP22a], which has the advantage of supporting efficient vector shortness proofs without slack. We use this feature to efficiently prove knowledge of, for example, a valid [DM14] signature.

Finally, we require a linking indistinguishable tag. We use a tag similar to those of [EBEK17, EKKS18], which can be seen as the lattice equivalent of DLOG-based tags mentioned above. To build a LIT tag $\mathbf{t}$ in the lattice setting, [EKKS18] use an LWE secret as the secret key, hash the message μ with the random oracle, and choose an error $\mathbf{e}$ to build an LWE sample from it, i.e. $\mathbf{t}^t = \mathbf{s}^t \cdot \mathcal{RO}(\mu) + \mathbf{e}^t$. Linking then works because if one tags the same message for the same secret key, the difference of the two tags is the difference of the two errors. Thus, the difference of two tags is short, iff they should link. [EKKS18] then show the security of their tag under the first-are-errorless

LWE assumption, a variant of LWE where the first few samples of an LWE oracle do not contain any error. Wen instanting the LIT, this costs them some efficiency, so we modify their construction to show our tag secure under the Module LWE assumption. We also introduce some new security notions for LITs in order to interface better with our generic construction.

Stateful reputation system. We also present a stateful variant of our generic construction of a reputation system in Section 6.1. The stateful variant works the same way as the stateless construction except for using stateful signatures as building blocks instead and having a fixed maximum number of users. The security proofs of the stateless construction can then easily be adapted to apply to the stateful variant. We then instantiate the stateful reputation system with two variants of the stateful signatures of [JRLS22], where the latter are described in Appendix C. Since the signatures of [JRLS22] are designed to be compatible with lattice-based proof systems and are more efficient than the signatures of [DM14], our signature constructions inherit this improved efficiency, which we then in turn improve the efficiency of our stateful reputation systems.

1.2. Related work

Reputation system constructions. Building reputation systems in the discrete logarithm setting is well-understood, with a wealth of papers with a variety of construction strategies and features. A good discussion can be found in the survey of Gurtler and Goldberg [GG21]. Closest to our generic construction are [BJK15, BEJ18], they are not quite instantiations of our generic construction, but they follow a similar paradigm (changes are mostly due to the fixed discrete logarithm setting in those papers, such as the usage of blind signatures to avoid signing public keys). Other papers, such as [LM19, BSS10], offer some form of privacy for issuers. In our construction, the issuer is known to all parties. We leave extensions, which offer some privacy to issuers, to future work and note that the techniques used here carry over to more complex scenarios. Another line of research considers reputation systems in a blockchain context, as surveyed by Hasan, Brunie, and Bertino [HBB23]. Those systems usually aim for trustlessness, i.e. ideally *no* party has to be trusted, but trust is distributed and backed by incentives throughout the blockchain network. Our system makes some trust assumptions, e.g., if group manager and issuer collude, we cannot prevent Sybil attacks. We do not model any reputation board party mentioned by [HBB23], which stores the rating signatures, but note that it can be realized by a public ledger, ensuring that ratings are not censored or deleted.

Lattice-based group signatures. One way to construct a reputation system is to take some group signature as base and then modify it such that linking is possible [EKKS18, BJK15, BEJ18]. This works because the notion of group signatures is closely related to anonymous reputation systems; one can view reputation systems as a group of group signatures. Both want to protect the anonymity of users inside a group or system, where the users authenticate messages, while a privileged opener is able to de-anonymize users. Therefore, we can explore existing lattice-based group signatures as potential bases for a lattice-based reputation system. One example is the group signature of [LNWX17], which [EKKS18] used to construct their reputation system, as explained later in more detail.

Another potential group signature to build a reputation system from is the one of [BCOS20], which uses the sign-encrypt-proof paradigm. They employ the Aurora SNARK [BSCR^{+}19] for their proofs, which has the advantage of no slack and very small proofs. However, the computation time for the proofs required by the group signature seems to be too high, as [BCOS20] explain.

In their paper on very efficient NIZKs with no slack, [LNPS21] also present a group signature scheme, which is based on the constructions of [DPLS18, LNPS21]. While this scheme promises very short signatures, their group signature is static, i.e. the group does not change. This does not match our dynamic model of a reputation system. Furthermore, [LNPS21] model their user identities as single ring elements of a special set, which they then sign to let the user join the group. However, in our construction we need to be able to sign the public keys of the LIT scheme, which generally do not fall into this special set.

Another group signature on which one could base a reputation system is the one by [LNWX18]. They also follow the sign-encrypt-proof paradigm, and concretely use the signatures of [DM14], an encryption scheme by [LPR13a] transformed to CCA security similar to the Naor-Yung paradigm and some Stern-like proof system. This group signature uses the same signature scheme and a similar encryption scheme as building blocks as we do in our first instantiation of the reputation system (note that we use different NIZKs).

Lattice-based reputation systems. To the best of our knowledge, the only other construction of a reputation system that is based on lattices is the construction of [EKKS18]. The idea for their construction is to start with the group signature from [LNWX17] and then view the reputation system as a group of group signatures. For each item that can be rated, the group manager sets up a separate group signature via a hash-based accumulator that is a Merkle-tree of all public keys of users who may rate the item. To create a rating a user then encrypts his identity, creates a tag with a LIT and then proves in zero-knowledge that he encrypted and tagged correctly as well as that his public key, for which he knows the secret key, is contained in the Merkle-tree.

A drawback of their model is that there are no issuers, instead there is a single group manager who manages everything. This gives the single group manager more power in a setting where there are different people to be rated, where these people need to trust the single group manager to work honestly. By separating the group manager from issuers, we can also split up their power, allowing for a more fine-grained approach of modelling trust. This is reflected in our security model. Additionally our security model offers a slightly stronger corruption model, except for requiring the opener to be honest (cf. Section 5.1).

Another drawback of the construction of [EKKS18] is that due to it relying on public Merkle-trees, there exists a public record of all users who can rate an item. While this does not contradict any formal security notion, in practice it is undesirable that the whole purchase history of all users is publicly available and a construction not exhibiting this issue is preferable. Our construction prevents this drawback by using signatures instead of a Merkle-tree to add users to the group. Obviously, even in our setting malicious issuers can always share the purchase history of users who bought from them with other people, but this is unpreventable. However, [EKKS18] *requires* their group manager to publish this information in order for the system to work. Furthermore, due to their usage of first-are-errorless LWE for the LIT as mentioned before and their usage of Stern-like proofs, the construction of [EKKS18] is less efficient than ours.

The advantage that the construction of [EKKS18] has over our construction is that they can assume the opener to be corrupt in every security notion but anonymity, while our construction needs the opener to be honest-but-curious. [EKKS18] achieve this requirement by introducing a Judge algorithm with which one can publicly verify that the opener worked correctly. We note that it is straight-forward to add Judge to our generic construction and our instantiations, but we omit it for better readability.

2. Preliminaries

We denote drawing some x uniformly from a set S by $x \leftarrow S$. We overload notation and denote by $x \leftarrow D$ sampling x from a distribution D. If $A(y)$ is a (probabilistic polynomial time (ppt)) algorithm, $x \leftarrow A(y)$ denotes sampling x from the output distribution of A on input y. We denote the random oracle as $\mathcal{RO}$.

We denote scalars as lowercase letters a, column vectors as bold lowercase letters $\mathbf{a}$ and matrices as bold uppercase letters $\mathbf{A}$. By $\mathbf{I}_c$ we denote the identity matrix of dimension $c \times c$. If the dimensions are clear from the context, we may only write $\mathbf{I}$. The same holds for $\mathbf{0}$, by which we denote the vector or matrix consisting of only zeroes. For the norm $\|\mathbf{a}\|$ of a vector we use the euclidian norm unless specified otherwise. We denote the infinity norm of a vector by $\|\mathbf{a}\|_\infty$.

Unless otherwise specified, let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ with $n \geq 16$ being a power of two and let $q > 16$ and $q = 3, 5 \mod 8$. Let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. We require such a q, because then $\mathcal{R}_q$ splits into $\mathcal{R}_q \cong \mathbb{F}_{q^{n/2}} \times \mathbb{F}_{q^{n/2}}$, where $\mathbb{F}_{q^{n/2}}$ denotes the field with $q^{n/2}$ elements, which we use for some results, e.g. Lemmas 2.4 and 2.7. We represent elements of $\mathcal{R}_q$ as vectors over $\mathbb{Z}_q^n$. In general, we use the coefficent embedding $\theta : \mathcal{R}_q \to \mathbb{Z}_q^n$, since for the $\mathcal{R}$ we use the canonical embedding is the same as the coefficent embedding up to a factor of $\sqrt{n}$ [JRLS22]. Define $\mathcal{R}_2 = \theta^{-1}(\{0,1\}^n)$ and $\mathcal{R}_{\pm 1} = \theta^{-1}(\{-1,0,1\}^n)$. By $\tilde{x}$ we refer to the constant coefficient of some polynomial $x \in \mathcal{R}$.

2.1. Lattices, Discrete Gaussians and Lattice Problems

Definition 2.1. *A (full-rank)* $\mathfrak{n}$*-dimensional lattice* Λ *is a discrete, additive subset of* $\mathbb{R}^\mathfrak{n}$. *It can be represented by a basis* $\mathbf{B} \in \mathbb{R}^{\mathfrak{n}\times\mathfrak{n}}$. *We then write* $\Lambda = \Lambda(\mathbf{B}) = \{\mathbf{z} : \mathbf{z} = \mathbf{B}\mathbf{x}, x \in$

$\mathbb{Z}^{\mathsf{n}}\}$. *For a matrix* $\mathbf{A} \in \mathbb{Z}_q^{\mathsf{n}\times\mathsf{m}}$, *define the q-ary lattice* $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^{\mathsf{m}} : \mathbf{A}\mathbf{x} = \mathbf{0} \mod q\}$.

Definition 2.2. *Define the Gaussian function* $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \left\|\mathbf{x} - \mathbf{c}\right\| / s^2)$. *Define the discrete Gaussian distribution* $D_{\Lambda+\mathbf{t},s,\mathbf{c}}$ *on a lattice coset* $\Lambda + \mathbf{t}$ *with center* $\mathbf{c}$ *and parameter* s *as*

$$D_{\Lambda+\mathbf{t},s,\mathbf{c}}(\mathbf{x}) = \begin{cases} \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda+\mathbf{t})} & \textit{if } \mathbf{x} \in \Lambda + \mathbf{t} \\ 0 & \textit{else} \end{cases}$$

For a ring $\mathcal{R}$, define the discrete Gaussian distribution $D_{\mathcal{R},s,c} = \theta^{-1}(D_{\theta(\mathcal{R}),s,\theta(c)})$. For both distributions we sometimes omit the center in the case $c = 0$.

The first lattice problem we base our security on is module learning with errors (MLWE).

Definition 2.3 (MLWE)**.** *Let* $q > 2$ *and* $k > 0$. *Let* $\mathcal{R}$ *be a ring and* $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. *Let* χ *be a distribution over* $\mathcal{R}_q$. *For a secret* $\mathbf{s} \in \mathcal{R}_q^k$, *the MLWE distribution is defined as choosing* $\mathbf{a} \leftarrow \mathcal{R}_q^k$ *and* $e \leftarrow \chi$, *computing* $b = \mathbf{s}^t\mathbf{a} + e \mod q$, *and outputting* $(\mathbf{a}, b)$.

The MLWE problem $\mathsf{MLWE}_{q,\mathcal{R},k,\chi}$ *is then defined as distinguishing between the MLWE distribution for a secret* $\mathbf{s} \leftarrow \mathcal{R}_q^k$ *and the uniform distribution over* $\mathcal{R}_q^{k+1}$. *For* $k = 1$, *this is equivalent to ring learning with errors (RLWE).*

It can be useful to group the $\mathbf{a}_i$ from m samples together as the column vectors of a matrix $\mathbf{A} \in \mathcal{R}_q^{k\times m}$ and the b_i as the entries of a vector $\mathbf{b} \in \mathcal{R}_q^m$, such that we have $\mathbf{s}^t\mathbf{A} + \mathbf{e}^t = \mathbf{b}^t$ for some error vector $\mathbf{e} \in \mathcal{R}_q^m$.

There exists an alternative version of the MLWE problem, where the secret is not sampled uniformly from $\mathcal{R}_q$, but instead sampled as $\mathbf{s} \leftarrow \chi^k$. This is called the normal form of MLWE.

The MLWE problems as described above are decisional problems. There exist computational variants, where the goal is to compute the secret $\mathbf{s}$, given samples from the MLWE distribution. This is called the (normal form) search MLWE problem $\mathsf{sMLWE}_{q,\mathcal{R},k,\chi}$.

In some cases, we need to set the parameters of the normal form MLWE problem in such a way that the secret used to create a set of m samples is unique, meaning that with overwhelming probability there is no other secret and error vector that could produce the samples. We show for what parameters this is the case.

Lemma 2.4 (Short MLWE secrets are unique)**.** *Let* $q \neq 2$ *be a prime with* $q = 3, 5 \mod 8$ *(or* $q = 1 \mod 2n$*),* $k > 0$, $n > 16$ *be a power of* 2, $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$. *Let* $B_\beta = \{e \in \mathcal{R}_q : \left\|e\right\|_\infty \leq \beta\}$. *Let* $\Delta \geq 0$ *such that* $2\beta + \Delta < q^{1/4}$. *Then there exists an* m *and a negligible function* negl *such that*

$$\Pr\left[\begin{array}{l} \exists(\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}') \in (B_\beta^k)^2 \times (B_\beta^m)^2 \\ \textit{with } \mathbf{s} \neq \mathbf{s}' \wedge \left\|\mathbf{b}\right\|_\infty \leq \Delta \end{array} : \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{k\times m} \\ \mathbf{b}^t = (\mathbf{s} - \mathbf{s}')^t\mathbf{A} + (\mathbf{e} - \mathbf{e}')^t \end{array}\right] \leq \mathsf{negl}(n).$$

The proof for this can be found in Appendix A.

The next two lattice problems we need are the ring and module variants of the short integer solution problem (SIS).

Definition 2.5 (RSIS). *Let $q > 2$ and $m, \beta > 0$. Let $\mathcal{R}$ be a ring and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. The RSIS problem $\mathsf{RSIS}_{q,\mathcal{R},m,\beta}$ is given a uniform vector $\mathbf{a} \leftarrow \mathcal{R}_q^m$ to find a non-trivial vector $\mathbf{x} \in \mathcal{R}_q^m \backslash \{\mathbf{0}\}$ such that $\mathbf{a}^t\mathbf{x} = \mathbf{0}$ and $\|\mathbf{x}\| \leq \beta$.*

Definition 2.6 (MSIS). *Let $q > 2$ and $d, m, \beta > 0$. Let $\mathcal{R}$ be a ring and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. The MSIS problem $\mathsf{MSIS}_{q,\mathcal{R},d,m,\beta}$ is given a uniform matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{d\times m}$ to find a non-trivial vector $\mathbf{x} \in \mathcal{R}_q^m \backslash \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} = \mathbf{0}$ and $\|\mathbf{x}\| \leq \beta$.*

For the MSIS problem there exists a so-called normal-form variant, where if $m > d$ the first d columns of $\mathbf{A}$ form the identity matrix. The probability that we can transform an $\mathbf{A}$ into its normal form can be analysed through the probability that a single uniform element in $\mathcal{R}_q$ is invertible.

Lemma 2.7. *Let n be a power of 2 and $q \geq 16$ a prime with $q = 3, 5 \mod 8$. For the ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ with $\mathcal{R} := \mathbb{Z}[X]/(X^n + 1)$ we have*

$$\eta := \Pr[a \textit{ not invertible} : a \leftarrow \mathcal{R}_q] \leq \frac{2}{q^{n/2}}.$$

The proof for this can be found in Appendix B. Then, we can combine this lemma together with Lemma B.3 to get the following corollary, which tells us with what probability we can transform $\mathbf{A}$ into its normal form.

Corollary 2.8. *Let n, q and ring $\mathcal{R}_q$ be as in the previous lemma. If matrix $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2], \mathbf{A}_1 \in \mathcal{R}_q^{k\times k}, \mathbf{A}_2 \in \mathcal{R}_q^{k\times(n-k)}$, is chosen uniformly at random from $\mathcal{R}_q^{k\times n}, n \geq k$, then with probability at least $1-4k\cdot q^{-n/2}$, there is a matrix $\mathbf{A}_2'$ such that for $\mathbf{A}' = [\mathbf{I}_k|\mathbf{A}_2']$*

$$\Lambda^\perp(\mathbf{A}) = \Lambda^\perp(\mathbf{A}').$$

2.2. Encryption Schemes

To construct our reputation system, we later need a CPA secure encryption scheme as a building block. For this, we consider a standard syntax definition.

Definition 2.9. *An encryption scheme Π consists of the following three ppt algorithms.*

- $\mathsf{KeyGen}(1^n)$*: The key generation algorithm on input a security parameter n outputs a tuple of a secret key and a public key* $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{Enc}(\mathsf{pk}, m)$*: The encryption algorithm on input a public key* pk *and a message m outputs a ciphertext c.*
- $\mathsf{Dec}(\mathsf{sk}, c)$*: The decryption algorithm on input a secret key* sk *and a ciphertext c outputs a message m.*

We say that Π *is correct, if for all security parameters* n*, all* $(\mathsf{sk}, \mathsf{pk})$ *output by* $\mathsf{KeyGen}(1^n)$*, and all messages* m*, it holds that* $\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m]$ *is overwhelming in* n*.*

We define CPA security with a standard definition as well.

```
$\mathsf{IND\text{-}CPA}_{\Pi,\mathcal{A},b}(n)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^n)$
$(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk})$
$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}(c)$
```

Definition 2.10. *We define the advantage of an adversary* $\mathcal{A}$ *against an encryption scheme* Π *as*

$$\mathsf{Adv}^{CPA}_{\Pi,\mathcal{A}}(n) = \left|\Pr[\mathsf{IND\text{-}CPA}_{\Pi,\mathcal{A},0}(n) = 1] - \Pr[\mathsf{IND\text{-}CPA}_{\Pi,\mathcal{A},1}](n) = 1\right|.$$

We say that the scheme Π *is IND-CPA secure if for all ppt adversaries* $\mathsf{Adv}^{CPA}_{\Pi,\mathcal{A}}(n)$ *is negligible.*

2.3. Signature Schemes

Another building block we need for the reputation system is an EUF-CMA secure signature scheme.

Definition 2.11 (Signature scheme)**.** *A signature scheme* Σ *consists of the following ppt algorithms:*

- $\mathsf{KeyGen}(1^n)$ *outputs secret key and public key pair* $(\mathsf{sk}, \mathsf{pk})$*.*
- $\mathsf{Sign}(\mathsf{sk}, m)$ *outputs signature* σ*.*
- $\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$ *is deterministic and outputs a bit.*

We say that Σ *is* correct *if for all* $n \in \mathbb{N}$*, all* $(\mathsf{sk}, \mathsf{pk})$ *output by* $\mathsf{KeyGen}(1^n)$*, all messages* m *in the message space (which is implicitly defined by the* pk*), and all* σ *output by* $\mathsf{Sign}(\mathsf{sk}, m)$*, we have* $\mathsf{Vrfy}(\mathsf{pk}, m, \sigma) = 1$*.*

The standard EUF-CMA security notion is then defined as follows.

Definition 2.12 (EUF-CMA)**.** *A signature scheme* Σ *is* existentially unforgeable under chosen-message attacks (EUF-CMA) *if for all ppt* $\mathcal{A}$*,*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{EUFCMA}}_{\Pi,\mathcal{A}}(n) = \Pr[&\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) = 1 \wedge \mathcal{A} \textit{ has not queried } m^* : \\ &(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^n), (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})] \leq \mathsf{negl}(n).\end{aligned}$$

2.4. NIZKs

We use non-interactive zero-knowledge proof systems in the random oracle model in this paper.

Definition 2.13 (NIZK)**.** *A non-interactive proof system (NIZK) for a relation* $\mathfrak{R}$ *in the random oracle model is defined as a triple* $\Pi_{\text{NIZK}} = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *of ppt algorithms:*

- $\mathsf{Setup}(1^n)$ *outputs a common reference string* crs*.*
- $\mathcal{P}^{\mathcal{RO}(\cdot)}(\mathsf{crs}, x, w, m)$ *given instance* x*, witness* w*, and a message* m*, outputs a proof* π*.*
- $\mathcal{V}^{\mathcal{RO}(\cdot)}(\mathsf{crs}, x, m, \pi)$ *outputs a bit* b*.*

To simplify notation, we sometimes omit the random oracle $\mathcal{RO}(\cdot)$*, but assume implicitly that the prover and verifier have access to it. We say that the NIZK is* correct*, if for all* $(x, w) \in \mathfrak{R}$ *and* $m \in \{0,1\}^*$*, we have that*

$$\Pr[\mathcal{V}(\mathsf{crs}, x, m, \mathcal{P}(\mathsf{crs}, x, w, m)) : \mathsf{crs} \leftarrow \mathsf{Setup}(1^n)] = 1.$$

For a relation $\mathfrak{R}$, $L_{\mathfrak{R}} = \{x \mid \exists w : \ (x, w) \in \mathfrak{R}\}$ is the *language* associated with $\mathfrak{R}$. The message m is additional data bound to the proof (e.g., including m in a Fiat-Shamir hash). Its role can be observed in Definition 2.17.

In order to display the relation $\mathfrak{R}$ that is proven, we will use the following notation for proofs.

Definition 2.14. *We denote the generation of a proof* $\pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w, m)$ *by*

$$\pi \leftarrow \text{NIZK}\{x; w; \mathfrak{R}(x, w)\}(m),$$

where $\mathcal{P}$ *is from a non-interactive proof system* Π_{NIZK} *for the relation* $\mathfrak{R}$*. We say "Verify* π*" to mean checking that* $\mathcal{V}(\mathsf{crs}, x, m, \pi) = 1$ *and we say "*π *verifies" or "*π *is valid" if* $\mathcal{V}(\mathsf{crs}, x, m, \pi) = 1$ *holds.*

With respect to security, we require the NIZK to be zero-knowledge (i.e. proofs can be simulated without a witness), sound (i.e. one cannot prove false statements), simulation-sound (i.e. one cannot prove false statements, even in the presence of simulated proofs), and straight-line extractable (i.e. there exists an extractor that can efficiently compute a witness from a valid proof without rewinding). These definitions are standard, we list them below, starting with zero-knowledge.

Definition 2.15 (Zero-Knowledge)**.** *A NIZK* Π *is* zero-knowledge *if there exists a simulator* $\mathcal{S}$ *consisting of three ppt algorithms* $\mathcal{S} = (\mathcal{S}.\mathsf{Setup}, \mathcal{S}.\mathcal{RO}, \mathcal{S}.\mathsf{Sim})$ *such that for all ppt* $\mathcal{A}$ *there exists a negligible function* negl *such that,*

$$\mathsf{Adv}^{ZK}_{\Pi,\mathcal{A}}(n) = \left| \begin{array}{rl} & \Pr[\mathcal{A}^{\mathcal{P}(\mathsf{crs},\cdot,\cdot,\cdot),\mathcal{RO}(\cdot)}(1^n, \mathsf{crs}) = 1 : \mathsf{crs} \leftarrow \mathsf{Setup}(1^n)] \\ - & \Pr[\mathcal{A}^{\mathsf{Sim}(\cdot,\cdot,\cdot),\mathcal{S}.\mathcal{RO}(\cdot)}(1^n, \mathsf{crs}) = 1 : \mathsf{crs} \leftarrow \mathcal{S}.\mathsf{Setup}(1^n)] \end{array} \right| \leq \mathsf{negl}(n)$$

where $\mathcal{RO}$ *denotes a random oracle. The oracle* $\mathsf{Sim}(x, w, m)$ *checks if* $(x, w) \in \mathfrak{R}$ *and if so, runs* $\mathcal{S}.\mathsf{Sim}(x, m)$*. We assume that* $\mathcal{S}$ *is stateful, i.e. it implicitly keeps state between invocations of* $\mathcal{S}.\mathsf{Setup}$*,* $\mathcal{S}.\mathcal{RO}$*, and* $\mathcal{S}.\mathsf{Sim}$*.*

We give the simulator two advantages beyond a regular prover that should allow it to efficiently simulate proofs without a witness: (1) $\mathcal{S}.\mathsf{Setup}$ generates crs and that process can yield a trapdoor that $\mathcal{S}$ stores in its state. (2) $\mathcal{S}$ answers the random oracle queries of $\mathcal{A}$ with $\mathcal{S}.\mathcal{RO}(\cdot)$, so $\mathcal{S}$ can program random oracle answers.

The second requirement we have is soundness, which states it is hard for an adversary to prove a false statement.

Definition 2.16 (Soundness)**.** *We say that a NIZK* Π *is sound if for all ppt* $\mathcal{A}$*, there is a negligible function* negl *such that*

$$\mathsf{Adv}^{Snd}_{\Pi,\mathcal{A}}(n) = \Pr\left[\begin{array}{l} \mathcal{V}^{\mathcal{RO}(\cdot)}(\mathsf{crs}, x, m, \pi) = 1 \land x \notin L_{\mathfrak{R}} : \\ \mathsf{crs} \leftarrow \mathsf{Setup}(1^n), (x, m, \pi) \leftarrow \mathcal{A}^{\mathcal{RO}(\cdot)}(1^n, \mathsf{crs}) \end{array}\right] \leq \mathsf{negl}(n)$$

Next, we require simulation soundness, i.e. even given access to an oracle creating simulated proofs (potentially for false statements), it is hard to compute an accepting proof for a wrong (not-queried) statement x.

Definition 2.17 (Simulation soundness)**.** *Let* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *be a zero-knowledge NIZK, with simulator* $\mathcal{S}$ *as in Definition 2.15. We say that* Π *is* simulation-sound *if for all ppt* $\mathcal{A}$*, there exists a negligible function* negl *such that*

$$\mathsf{Adv}^{SS}_{\Pi,\mathcal{A}}(n) =$$
$$\Pr\left[\begin{array}{ll} & \mathcal{V}^{\mathcal{S}.\mathcal{RO}(\cdot)}(\mathsf{crs}, x, m, \pi) = 1 \\ \land & x \notin L_{\mathfrak{R}} \\ \land & \mathcal{A} \textit{ has not queried } \mathcal{S}.\mathsf{Sim}(x, m) \end{array} : \begin{array}{l} \mathsf{crs} \leftarrow \mathcal{S}.\mathsf{Setup}(1^n), \\ (x, m, \pi) \leftarrow \mathcal{A}^{\mathcal{S}.\mathsf{Sim}(\cdot,\cdot), \mathcal{S}.\mathcal{RO}(\cdot)}(1^n, \mathsf{crs}) \end{array}\right]$$
$$\leq \mathsf{negl}(n)$$

Note that, as usual, $\mathcal{A}$ may even query $\mathcal{S}.\mathsf{Sim}(x, m)$ for $x \notin L$. The simulation soundness property is sometimes understood to imply non-malleability of the proof π, i.e. defined with the condition "π has not been output by $\mathcal{S}.\mathsf{Sim}(x, m)$" instead of "$\mathcal{A}$ has not queried $\mathcal{S}.\mathsf{Sim}(x, m)$". We use the weaker condition here, which corresponds to the fact that we do not consider immaterial changes to rating signatures (e.g., re-randomization with no change to the rating text or the rated party) an attack (see, for example, Definition 5.4).

Finally, we require straight-line extractability.

Definition 2.18 (Straight-line extractability)**.** *Let* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *be a NIZK. We say that* Π *is a* straight-line extractable proof of knowledge *if there are ppt algorithms* $\mathcal{E}_0, \mathcal{E}_1$ *such that for all ppt* $\mathcal{A}_0, \mathcal{A}_1$*, there exist negligible functions* $\mathsf{negl}_0, \mathsf{negl}_1$ *such that*

$$\mathsf{Adv}^{\mathrm{PoK}_0}_{\Pi,\mathcal{A}_0}(n) = \left|\begin{array}{ll} & \Pr[\mathcal{A}_0(1^n, \mathsf{crs}) = 1 : \mathsf{crs} \leftarrow \mathsf{Setup}(1^n)] \\ - & \Pr[\mathcal{A}_0(1^n, \mathsf{crs}) = 1 : (\mathsf{crs}, td) \leftarrow \mathcal{E}_0(1^n)] \end{array}\right| \leq \mathsf{negl}_0(n)$$

and

$$\mathsf{Adv}^{\mathrm{PoK}_1}_{\Pi,\mathcal{A}_1}(n) = \Pr\left[\begin{array}{ll} & \mathcal{V}^{\mathcal{RO}}(\mathsf{crs}, x, m, \pi) = 1 \\ \land & (x, w) \notin \mathfrak{R} \end{array} : \begin{array}{l} (\mathsf{crs}, td) \leftarrow \mathcal{E}_0(1^n), \\ (x, m, \pi) \leftarrow \mathcal{A}_1(1^n, \mathsf{crs}), \\ w \leftarrow \mathcal{E}_1(td, x, m, \pi) \end{array}\right] \leq \mathsf{negl}_1(n)$$

In the random oracle model, $\mathcal{E}_1$ gets the list of random oracle queries that $\mathcal{A}$ made as additional input.

We give the extractor the advantage of setting up crs (allowing it to embed a trapdoor td) and, in the random oracle model, of observing the random oracle queries of $\mathcal{A}$ (as in [Fis05a]). The extractor does not have any ability to rewind $\mathcal{A}$, so extraction through rewinding is not an option. Note that in this security definition, we do not give $\mathcal{A}$ access to simulated proofs.

3. Linking Indistinguishable Tags

A building block we need are *linking indistinguishable tags* (LIT). The idea of such a scheme is that one can compute a tag for a given message with a secret key. An adversary should not able to tell which secret key was used to create the tag. However, if one tags the same message twice, i.e. with the same secret key, anyone can discover this by linking the tags. There also exists a function f from which we can compute a public key $\mathsf{pk} = f(\mathsf{sk})$. We typically require f to be a one-way function implicitly. This public key is not used in the scheme itself, but can be used in conjunction with other primitives. The formal model looks as follows.

Definition 3.1. *A linking indistinguishable tags scheme consists of a function f and the following ppt algorithms:*

- $\mathsf{KeyGen}(1^n)$*: On input a security parameter n, it outputs a secret* sk.
- $\mathsf{Tag}(\mathsf{sk}, \mu)$*: On input a secret key* sk *and a message μ, it outputs a tag t.*
- $\mathsf{Vrfy}(\mathsf{sk}, \mu, t)$*: On input a secret key* sk*, a message μ and a tag t, it outputs a bit b.*
- $\mathsf{Link}(\mu, t_0, t_1)$*: On input a message μ and two tags t_0, t_1, it outputs a bit b.*

We require that a LIT is correct. This is the case if for all security parameters n, all sk *output by* $\mathsf{KeyGen}(1^n)$*, all messages μ, all tags t_0, t_1 output by* $\mathsf{Tag}(\mathsf{sk}, \mu)$*, we have that* $\mathsf{Vrfy}(\mathsf{sk}, \mu, t_0) = 1$ *and* $\mathsf{Link}(\mu, t_0, t_1) = 1$*.*

The first security requirement is tag-indistinguishability. In this standard indistinguishability game an adversary has to decide which of two secrets was used to create the challenge, while having access to tag oracle for these secrets. We define the oracle $\mathsf{Tg}(c, \mu)$ to return t if there exists some $(c, \mu, t) \in Q$. Else, we return $t \leftarrow \mathsf{Tag}(\mathsf{sk}_c, \mu)$ and add (c, μ, t) to Q.

```
$\mathsf{Anon}^{LIT}_{\Pi,\mathcal{A},b}(n)$
$\mathsf{sk}_0, \mathsf{sk}_1 \leftarrow \mathsf{KeyGen}(1^n)$
$\mathsf{pk}_i = f(\mathsf{sk}_i), i \in \{0,1\}$
$\mu^* \leftarrow \mathcal{A}^{\mathsf{Tg}(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$t^* \leftarrow \mathsf{Tag}(\mathsf{sk}_b, \mu^*)$
$b' \leftarrow \mathcal{A}^{\mathsf{Tg}(\cdot,\cdot)}(t^*)$
If $\mu^*$ was queried, output 0, else output $b'$.
```

Definition 3.2. *A LIT* Π *has tag-indistinguishability, if there exists a negligible function* negl *such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{LITAnon}_{\Pi,\mathcal{A}}(n) := \left|\Pr[\mathsf{Anon}^{LIT}_{\Pi,\mathcal{A},0}(n) = 1] - \Pr[\mathsf{Anon}^{LIT}_{\pi,\mathcal{A},1}(n) = 1]\right| \leq \mathsf{negl}(n).$$

The second security requirement is linkability. This asks that no adversary can produce two secret key tag pairs and a message, such that the secret key tag pairs are valid for the message, while the tags do not link. In comparison to the security model of [EKKS18], we generalize our security model for linkability and allow the adversary to output two different secret keys, but they must map to the same public key.

```
$\mathsf{Linkable}^{LIT}_{\Pi,\mathcal{A}}(n)$
$(\mathsf{sk}_0, \mathsf{sk}_1, \mu, t_0, t_1) \leftarrow \mathcal{A}(1^n)$
If $f(\mathsf{sk}_0) \neq f(\mathsf{sk}_1)$ or $\exists i \in \{0,1\} : \mathsf{Vrfy}(\mathsf{sk}_i, \mu, t_i) = 0$, return 0.
If $\mathsf{Link}(\mu, t_0, t_1) = 0$, output 1.
```

Definition 3.3. *A LIT* Π *has linkability if there exists a negligible function* negl *such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{Linkable}^{LIT}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

Another security requirement, unforgeability, is similar to the requirement for a one-way function. It requires that no adversary is able to produce a secret key, message and valid tag, such that the tag links to another valid tag. For that, we need a tag oracle QTg, that on input (sk, μ) returns t if there exists $(\mu, t) \in Q$. Else it computes $t \leftarrow \mathsf{Tag}(\mathsf{sk}, \mu)$, adds (μ, t) to $\mathcal{Q}$ and returns t.

```
$\mathsf{Forge}^{LIT}_{\Pi,\mathcal{A}}(n)$
$\mathcal{Q} = \emptyset$
$\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^n), \mathsf{pk} = f(\mathsf{sk})$
$(\mathsf{sk}^*, \mu, t^*) \leftarrow \mathcal{A}^{\mathsf{QTg}(\mathsf{sk},\cdot)}(\mathsf{pk})$
If $\mathsf{Vrfy}(\mathsf{sk}^*, \mu, t^*) = 0$, output 0.
If $\exists\ (\mu, t) \in \mathcal{Q}$ such that $\mathsf{Link}(\mu, t, t^*) = 1$, output 1.
```

Definition 3.4. *A LIT Π is unforgeable, if there exists a negligible function* negl *such that for all ppt adversaries $\mathcal{A}$ it holds that*

$$\Pr[\mathsf{Forge}^{LIT}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

The last requirement for LIT schemes is non-invertability. This asks that an adversary is not able to find a secret key to given public key, while having access to a tag oracle.

```
$\mathsf{Invert}_{\Pi,\mathcal{A}}(n)$
$\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^n)$
$\mathsf{pk} = f(\mathsf{sk})$
$\mathsf{sk}' \leftarrow \mathcal{A}^{\mathsf{QTg}(\mathsf{sk},\cdot)}(\mathsf{pk})$
If $\mathsf{pk} = f(\mathsf{sk}')$, output 1.
```

Here QTg is defined as before.

Definition 3.5. *A LIT Π has non-invertability, if there exists a negligible function* negl *such that for all ppt adversaries $\mathcal{A}$ it holds that*

$$\Pr[\mathsf{Invert}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

3.1. Construction Based on Module Lattices

Given the formal model of a LIT, we now want to construct a LIT based on module lattice problems. One can also construct a similar LIT based on unstructured lattice problems, but we focus on the module case with module rank k. When we later use the LIT in our reputation system, we only need $k = 1$, in which case the security assumption for the LIT reduces to ideal lattices. We think the LIT may be of independent interest, so we construct it with general rank k.

The idea for the construction is that a public key is simply a batch of MLWE samples for some secret $\mathbf{s}$. A tag on a message μ then is the second component of another batch of MLWE samples, i.e. $\mathbf{t}^t = \mathbf{s}^t\mathbf{A}_\mu + \mathbf{e}'^t$, for the same secret $\mathbf{s}$ and some different error $\mathbf{e}'$, where we define $\mathbf{A}_\mu = \mathcal{RO}(\mu)$. This way, if we tag the same message twice, the $\mathbf{A}_\mu$ is the same for both tags, and the difference of the two tags is equal to the difference of the two errors. Since this is short, we can detect that the tags were created for the same message.

Construction 3.6. Let $m, k > 0$. Let $\beta < 2^{-\frac{n}{mk}+\frac{n}{2k}\log(q)-3}$. Let χ be a distribution over $\mathcal{R}_q$. Construct the LIT Π_{LIT} consisting of the following algorithms:

- $\mathsf{KeyGen}(1^n)$: Choose $\mathbf{s} \leftarrow \chi^k, \mathbf{e} \leftarrow \chi^m$. Set $\mathsf{sk} = (\mathbf{s}, \mathbf{e})$.
- $\mathsf{Tag}(\mathsf{sk}, \mu)$: Compute $\mathbf{A}_\mu = \mathcal{RO}(\mu) \in \mathcal{R}_q^{k\times m}$ and $\mathbf{e}' \leftarrow \chi^m$. Output $\mathbf{t}^t = \mathbf{s}^t\mathbf{A}_\mu + \mathbf{e}'^t$.
- $\mathsf{Vrfy}(\mathsf{sk}, \mu, \mathbf{t})$: Compute $\mathbf{A}_\mu = \mathcal{RO}(\mu) \in \mathcal{R}_q^{k\times m}$. If $\|\mathbf{t} - (\mathbf{s}^t\mathbf{A}_\mu)^t\|_\infty < \beta$ and $\|\mathbf{s}\|_\infty \leq \beta$, output 1.

- $\mathsf{Link}(\mu, \mathbf{t}_0, \mathbf{t}_1)$: If $\|\mathbf{t}_0 - \mathbf{t}_1\|_\infty < 2\beta$, output 1.
- $f = f_{\mathbf{A}}$ for $\mathbf{A} \leftarrow \mathcal{R}_q^{k\times m}$, $f_{\mathbf{A}}(\mathsf{sk}) = (\mathbf{s}^t\mathbf{A} + \mathbf{e}^t)^t$

Lemma 3.7. *The LIT Π_{LIT} has tag-indistinguishability (Definition 3.2) in the random oracle model, if normal form $\mathsf{MLWE}_{q,\mathcal{R},k,\chi}$ is hard.*

This can be proven by proving that $\mathsf{Anon}^{LIT}_{\Pi_{\mathsf{LIT}},\mathcal{A},0}(n)$ is indistinguishable from a game where the challenge tag t^* is generated uniformly at random, which is possible using the indistinguishability of the MLWE distribution from the uniform distribution. Then, one does the same for $\mathsf{Anon}^{LIT}_{\Pi_{\mathsf{LIT}},\mathcal{A},1}(n)$, from which we can see that the two games are indistinguishable if normal form MLWE is hard.

Lemma 3.8. *The LIT Π_{LIT} has non-invertability (Definition 3.5) in the random oracle model if normal form $\mathsf{sMLWE}_{q,\mathcal{R},k,\chi}$ is hard.*

Proof. Let $\mathcal{A}$ be an adversary against the invertability of the LIT. We construct an adversary $\mathcal{B}$ against normal form search-MLWE from it. $\mathcal{B}$ simulates $\mathcal{A}$ by using batching m samples from his MLWE oracle into a public key pk. By the definition of the MLWE oracle, there is some secret $\mathbf{s}$ that was used to generated these samples. When $\mathcal{A}$ asks for a tag on a previously unqueried message μ, $\mathcal{B}$ uses its MLWE oracle to get a batch of m samples $(\mathbf{A}, \mathbf{b})$, defines $\mathcal{RO}(\mu) := \mathbf{A}$ and answers with $\mathbf{b}$. If $\mathcal{A}$ asks for a tag on a previously queried μ, $\mathcal{B}$ answers with the $\mathbf{b}$ it generated before. When $\mathcal{A}$ outputs some $\mathsf{sk}' = (\mathbf{s}', \mathbf{e}')$, $\mathcal{B}$ returns $\mathbf{s}'$ to its challenger.

Due to Lemma 2.4 we know that the secret $\mathbf{s}$ behind the tags is unique, therefore we know $\mathbf{s} = \mathbf{s}'$ if $\mathcal{A}$ wins and thus $\mathbf{s}'$ is a valid solution for normal form search-MLWE. □

Lemma 3.9. *The LIT Π_{LIT} is linkable (Definition 3.3) in the random oracle model.*

Proof. The adversary can only win, if $f(\mathsf{sk}_0) = f(\mathsf{sk}_1)$. This means, that $\mathbf{s}_0^t\mathbf{A} + \mathbf{e}_0^t = \mathbf{s}_1^t\mathbf{A} + \mathbf{e}_1^t$, where $\mathsf{sk}_i = (\mathbf{s}_i, \mathbf{e}_i)$. Due to Lemma 2.4 we know that the short MLWE secrets are unique, meaning $\mathbf{s}_0 = \mathbf{s}_1$. Therefore we know that $\mathbf{t}_0 - \mathbf{t}_1 = \mathbf{s}_0^t\mathbf{A}_\mu + \mathbf{e}_0'^t - \mathbf{s}_1^t\mathbf{A}_\mu - \mathbf{e}_1'^t = \mathbf{e}_0'^t - \mathbf{e}_1'^t$ for some $\mathbf{e}_i', i \in \{0,1\}$ with $\|\mathbf{e}_i'\|_\infty \leq \beta$. Thus we have $\|\mathbf{t}_0 - \mathbf{t}_1\|_\infty \leq 2\beta$ which is why the Link algorithm always outputs 1, meaning an adversary cannot win the linking game. □

Lemma 3.10. *The LIT Π_{LIT} is unforgeable (Definition 3.4) in the random oracle model if normal form $\mathsf{sMLWE}_{q,\mathcal{R},k,\chi}$ is hard.*

Proof. Let $\mathcal{A}$ be an adversary against the unforgeability of the LIT and let Q be the number of oracle queries of $\mathcal{A}$. Construct an adversary $\mathcal{B}$ against normal form search-MLWE. $\mathcal{B}$ uses the first m samples of its oracle as the pk and gives that to $\mathcal{A}$. Then, on tag-query μ, $\mathcal{B}$ asks its oracle for m samples batched as $(\mathbf{A}, \mathbf{b})$, programs the random oracle as $\mathcal{RO}(\mu) := \mathbf{A}$ and returns $\mathbf{b}$. This way, there is a consistent $\mathbf{s}$ behind the pk and tags $\mathcal{A}$ sees, although $\mathcal{B}$ does not know it. $\mathcal{A}$ then outputs some sk^*, μ and t^*. If the tag is valid and links to some tag t, $\mathcal{B}$ outputs $\mathbf{s}^*$, where $\mathsf{sk}^* = (\mathbf{s}^*, \cdot)$.

Now, due to Lemma 2.4 and the choice of β we know that the probability that $\mathbf{s} \neq \mathbf{s}^*$ is negligible. Therefore, if $\mathcal{A}$ finds a forgery, $\mathcal{B}$ outputs a solution for normal form search-MWLE with overwhelming probability.

□

4. Building Blocks

4.1. Lattice-Based Signatures

One way to construct (lattice) signatures is with the help of some trapdoors, for example the signature of [DM14]. Thus, we first define what a G-trapdoor for lattices is and how we generate one.

Definition 4.1 (G-trapdoor [MP12]). *For a matrix $\mathbf{a}^t \in \mathcal{R}_q^{1\times m}$, a G-trapdoor is a matrix $\mathbf{R} \in \mathcal{R}_q^{m\times\zeta}$ such that $\mathbf{a}^t\mathbf{R} = \mathbf{g}^t$ for a gadget matrix $\mathbf{g}^t \in \mathcal{R}_q^{1\times\zeta}$.*

We can generate such trapdoors with an algorithm called $\mathsf{GenTrap}$ and use them to sample preimages of some function for a given image with $\mathsf{PreSample}$.

Theorem 4.2 ([MP12]). *Let $\zeta \in \mathbb{N}$ and $m = \mathcal{O}(n \log q)$ large enough. Let $g = \lceil q^{1/\zeta} \rfloor \in \mathcal{R}_q$ and $\mathbf{g}^t = [1 \mid g \mid \ldots \mid g^{\zeta-1}]$. There exist ppt algorithms $\mathsf{GenTrap}, \mathsf{PreSample}$ such that*

- $\mathsf{GenTrap}(1^n, 1^m, q)$ *outputs $\mathbf{a}^t \in \mathcal{R}_q^{1\times 2m}$ and $\mathbf{R} \in \mathcal{R}_q^{2m\times\zeta}$ such that $\mathbf{a}^t\mathbf{R} = \mathbf{g}^t$ and $\mathbf{R} \in \mathcal{R}_q^{2m\times\zeta}$ and the distribution of $\mathbf{a}^t$ is statistically indistinguishable from uniform;*
- $\mathsf{PreSample}(\mathbf{a}^t, \mathbf{R}, u, s)$ *on input a matrix $\mathbf{a}^t \in \mathcal{R}_q^{1\times m}$, a matrix $\mathbf{R} \in \mathcal{R}_q^{2m\times\zeta}$ output by* $\mathsf{GenTrap}$, *a syndrome $u \in \mathcal{R}_q$ and a standard deviation $s \geq \eta_\epsilon(\mathbb{Z})\sqrt{g^2+1}\sqrt{\|\mathbf{R}\|^2}$ outputs $\mathbf{v}$ that is statistically close to $D_{\mathcal{R}_q^{2m},s}$ conditioned on $\mathbf{a}^t\mathbf{v} = u \mod q$.*

We now state the signature of [DM14] that we later to use to instantiate our reputation system. We claim that it is (adaptively) EUF-CMA secure without any changes, while the original theorem by [DM14] only claims security for non-adaptive queries. [DM14] achieve adaptive security by a standard transformation of first hashing the message with a chameleon hash before signing it. However, one can adapt their security proof to directly show adaptive security by applying a technique similar to [LSS14] using the Rényi divergence.

The idea of the construction of [DM14] is that the public key contains some uniformly generated $\mathbf{a}^t$, while the secret key is a trapdoor for that $\mathbf{a}$. To sign a message $\mathbf{m}$, we first choose a random tag κ. Based on $\kappa, \mathbf{a}^t$ and some public matrices $\mathbf{a}_i^t$ we then define some $\mathbf{a}_\kappa^t$ in such a way that we can adapt the trapdoor for $\mathbf{a}^t$ to a trapdoor for $\mathbf{a}_\kappa^t$. We then hash the message using some $\mathbf{d}^t$ and add some public u to get $v = u + \mathbf{d}^t\mathbf{m}$. Then, we use $\mathsf{PreSample}$ to sample a short preimage σ of v under $\mathbf{a}_\kappa^t$. Signature verification is then simply checking whether σ is indeed a short preimage of v and whether κ is in the tag space.

Construction 4.3. Let the message space be $\mathcal{R}_2^{m_2}$. Let $g = \lceil q^{\frac{1}{\zeta}} \rfloor$ and $\mathbf{g}^t = [1 \mid g \mid \ldots \mid g^{\zeta-1}] \in \mathcal{R}_q^{1\times\zeta}$. Let the tag space be $\mathcal{T} = \{0,1\}^d$. Let $s = n^{3/2} \cdot \omega(\log n)^{3/2}$ such that $s^2 \geq (\sqrt{nm_1} + \sqrt{nm_2} + t)\sqrt{nm_2}$ for some t. Let $\beta = s\sqrt{n(m_1+\zeta)}$.

- $\mathsf{KeyGen}(1^n)$: Choose $(\mathbf{a}^t, \mathbf{R}) \leftarrow \mathsf{GenTrap}(1^n, 1^{m_1}, q)$ such that $\mathbf{a}^t \in \mathcal{R}_q^{1\times m_1}, \mathbf{R} \in \mathcal{R}_q^{m_1\times\zeta}$ and $\mathbf{a}^t\mathbf{R} = \mathbf{g}^t$. Choose $\mathbf{a}_i^t \leftarrow \mathcal{R}_q^{1\times m_1}$ for $i \in \{0,\ldots,d\}$. Choose $\mathbf{d}^t \leftarrow \mathcal{R}_q^{1\times m_2}$, $u \leftarrow \mathcal{R}_q$. Set $\mathsf{pk} = (\mathbf{a}^t, \mathbf{a}_0^t, \ldots, \mathbf{a}_d^t, \mathbf{d}^t, u)$ and $\mathsf{sk} = \mathbf{R}$.

- $\mathsf{Sign}(\mathsf{sk}, \mathbf{m})$: Choose $\kappa \leftarrow \mathcal{T}$. Set $\mathbf{a}_\kappa^t = [\mathbf{a}^t \mid \mathbf{a}_0^t + \sum_{i=1}^{d} \kappa_i \mathbf{a}_i^t]$, where κ_i denotes the ith bit of κ. Compute $\sigma \leftarrow \mathsf{PreSample}(\mathbf{a}_\kappa^t, (\mathbf{R}^t, \mathbf{0})^t, u + \mathbf{d}^t\mathbf{m}, s)$. Output (κ, σ).
- $\mathsf{Vrfy}(\mathsf{pk}, \mathbf{m}, (\kappa, \sigma))$: If $\mathbf{a}_\kappa^t \sigma = u + \mathbf{d}^t\mathbf{m}$ and $\|\sigma\| \leq \beta$ and $\kappa \in \mathcal{T}$, output 1.

We claim security of the signature scheme as follows.

Theorem 4.4. *For every ppt adversary $\mathcal{A}$ that makes at most $Q \leq 2^{o(n)}$ signature queries and has EUF-CMA advantage ϵ, there exists an adversary $\mathcal{B}$ against $\mathsf{RSIS}_{\mathcal{R}_q,m,q,\beta'}$ with advantage $\left(\frac{\epsilon}{4Q^2}\right)^c (\epsilon(n)/2 - \mathsf{negl}(n))^{\alpha/(\alpha-1)} \cdot \exp(-\pi\alpha) - 2^{-\Omega(n)}$, where $\beta' = n^{7/2} \cdot \log n \cdot \omega(\log n)^{5/2}$, for any $\alpha > 1$.*

We now describe how one can change the proof of [DM14] in order to get adaptive security directly. In their proof, [DM14] can already answer all signature queries adaptively, except for at most one, since they know a trapdoor for the corresponding $\mathbf{a}_\kappa^t$. Only if there exists a query j such that $\kappa_{\leq i^*}^{(j)} = \kappa_{\leq i^*}^*$, i.e. if there exists a query j where the i^* bit long prefix of the jth tag $\kappa^{(j)}$ is the same as the guessed prefix $\kappa_{\leq i^*}^*$, there is no trapdoor. Thus [DM14] generate the signature answer σ^* not with a trapdoor but through other means, for which they need the non-adaptiveness. We change how we generate σ^* in this case and some other public values and then analyse the changes. In the beginning, when given an RSIS instance $\mathbf{a}^t \in \mathcal{R}_q^{1 \times m_1}$, we choose the tags $\kappa^{(1)}, \ldots, \kappa^{(Q)}$ to be used in the signature queries. Then, we generate $\mathbf{d}^t$ by choosing $\mathbf{U} \leftarrow \mathcal{R}_{\pm 1}^{m_1 \times m_2}$ and setting $\mathbf{d}^t = \mathbf{a}^t\mathbf{U}$. We define an index i^* and guess a prefix $\kappa_{\leq i^*}^* \leftarrow T_{i^*}$ as in [DM14]. Furthermore, if a j exists such that $\kappa_{\leq i^*}^{(j)} = \kappa_{\leq i^*}^*$, we generate u by choosing $\mathbf{e} \leftarrow D_{\mathcal{R},s}^{m_1+\zeta}$ and setting $u = \mathbf{a}_{\kappa^{(j)}}^t \mathbf{e}$. Then, in the jth query, if $\kappa_{\leq i^*}^{(j)} = \kappa_{\leq i^*}^*$ holds, we answer with $\mathbf{e}' = \mathbf{e} + \mathbf{d}$, where $\mathbf{d} = \begin{bmatrix} \mathbf{Um} \\ \mathbf{0} \end{bmatrix}$. If no such j exists, we proceed as in [DM14], so we only look at the case where such a j exists.

We now want to argue that these changes are indistinguishable to a ppt adversary. From [BJRLW23, Lemma 2.8] we know that the distribution of $(\mathbf{a}^t, \mathbf{a}^t\mathbf{U})$ is statistically close to uniform. Due to Corollary 2.8 we know that there exists a transformation of $\mathbf{a}^t$ to its normal form with probability $1 - 4q^{n/2}$. Then with Corollary 7.4 of [LPR13a, LPR13b] we know that $(\mathbf{a}^t, \mathbf{a}^t\mathbf{e})$ is statistically close to uniform since $s > 2nq^{1/m_1 + 2/(nm_1)}$. Thus, we lastly have to argue about the distribution of $\mathbf{e}'$. We will argue that the distribution of $\mathbf{e}'$ is statistically indistinguishable from the signature in the real game, when both are conditioned on the respective values of $\mathbf{a}^t, \mathbf{d}^t, u$. Then we know that the joint distribution of pk together with $\mathbf{e}'$ is statistically indistinguishable. Let $\mathbf{z}$ be some solution to $\mathbf{a}_{\kappa^{(j)}}^t \mathbf{z} = \mathbf{u}$. Then, we know that in the original game the distribution of the jth signature σ_j conditioned on pk is $D_{\Lambda^\perp(\mathbf{a}_{\kappa^{(j)}}^t)+\mathbf{z}+\mathbf{d},s} = D_{\Lambda^\perp(\mathbf{a}_{\kappa^{(j)}}^t),s,-\mathbf{z}-\mathbf{d}} + \mathbf{z} + \mathbf{d}$. If we look at the distribution of $\mathbf{e}$ conditioned on the pk generated by $\mathcal{B}$, we see that its distribution is $D_{\Lambda^\perp(\mathbf{a}_{\kappa^{(j)}}^t)+\mathbf{z},s}$. Thus, the distribution of $\mathbf{e}'$ is $D_{\Lambda^\perp(\mathbf{a}_{\kappa^{(j)}}^t)+\mathbf{z},s} + \mathbf{d} = D_{\Lambda^\perp(\mathbf{a}_{\kappa^{(j)}}^t),s,-\mathbf{z}} + \mathbf{z} + \mathbf{d}$ and the distributions of σ_j and $\mathbf{e}'$ only differ in their center. Therefore, from [LSS14, Lemma 4.2] we know that the Rényi difference of the two distributions is smaller than

$\exp(\alpha\pi \left\|\mathbf{d}\right\|_2^2 / s^2) \leq \exp(\alpha\pi)$ for any $\alpha > 0$, where the latter holds by construction. Then it holds that $\Pr[W_{i^*}]^{\alpha/(\alpha-1)} \leq \exp(\alpha\pi \left\|\mathbf{d}\right\|^2 / s^2) \Pr[W_{\mathbf{e}'}]$ by [LSS14, Lemma 4.1], where W_{i^*} is the event that the signature adversary outputs a valid forgery when given σ_{i^*} in the simulation (with changed public keys) and $W_{\mathbf{e}'}$ is the event that the signature adversary outputs a valid forgery when given $\mathbf{e}'$. The rest of the proof works as in [DM14]. Thus, together with the analysis from [DM14] we know that the probability $\gamma(n)$ that the RSIS adversary outputs a valid solution is

$$\begin{aligned}\gamma(n) &\geq \frac{1}{|T_{i^*}|} \left(\epsilon(n)/2 - \mathsf{negl}(n)\right)^{\alpha/(\alpha-1)} \cdot \exp(-\pi\alpha) - 2^{-\Omega(n)} \\ &\geq \left(\frac{\epsilon}{4Q^2}\right)^c \left(\epsilon(n)/2 - \mathsf{negl}(n)\right)^{\alpha/(\alpha-1)} \cdot \exp(-\pi\alpha) - 2^{-\Omega(n)},\end{aligned}$$

where negl is a negligible function and c is defined as in [DM14]. Thus, we get adaptive security without having to use chameleon hashes at the cost of some reduction loss introduced by the Rényi divergence.

4.2. Lattice Encryption

We describe a module variant of the encryption scheme presented in [Reg09]. To encrypt more than one ring element, we increase the size of the secret key instead of encrypting each ring element separately, thus we keep the size of the ciphertext smaller than in the other case. This is beneficial if we later want to prove that we encrypted a ciphertext honestly.

Construction 4.5. Let $k, m' > 0$ and $m \geq k + m'$. Let $s > 0$ and $s' > 2n \cdot q^{(k+m')/m + 2/(nm)}$. Let $\chi = D_{\mathcal{R},s}$ and $\chi' = D_{\mathcal{R},s'}$.

- $\mathsf{KeyGen}(1^n)$: Choose $\bar{\mathbf{A}} \leftarrow \mathcal{R}_q^{k \times m}$. Choose $\mathbf{S} \leftarrow \chi^{m' \times k}$ and $\mathbf{E} \leftarrow \chi^{m' \times m}$ and set $\mathsf{sk} = \mathbf{S}$. Set $\mathbf{B} = \mathbf{S}\bar{\mathbf{A}} + \mathbf{E}$. Set $\mathsf{pk} = \mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{B} \end{bmatrix}$.
- $\mathsf{Enc}(\mathsf{pk}, \mu \in \{0,1\}^{nm'})$: Interpret μ as a vector $\mathbf{m} \in \mathcal{R}_q^{m'}$. Choose $\mathbf{r} \leftarrow \chi'^m$. Set $\mathbf{c} = \mathbf{A}\mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor q/2 \rceil \cdot \mathbf{m} \end{bmatrix}$. Output $\mathbf{c}$.
- $\mathsf{Dec}(\mathsf{sk}, \mathbf{C})$: Compute $\mathbf{m}' = \begin{bmatrix} -\mathbf{S} \mid \mathbf{I} \end{bmatrix} \mathbf{c}$, interpret $\mathbf{m}'$ as a vector of $\mathbb{Z}_q$ elements and check entrywise whether the entry is closer to 0 or $q/2$ to get back μ.

The CPA security of this scheme can be shown if s' is large enough such that $\mathbf{A}\mathbf{r}$ is statistically indistinguishable from uniform and if one assumes that $\mathsf{MLWE}_{q,\mathcal{R},k,m,\chi}$ is hard. This is done by first game-hopping to a game, where the public key is chosen uniformly at random instead, which is indistinguishable by the aforementioned assumption. Then, one can argue with Corollary 2.8 that one can transform $\mathbf{A}$ into its normal form with overwhelming probability, so that one can use Corollary 7.4 of [LPR13a, LPR13b] to show that the challenge ciphertext is indistinguishable from uniformly random, if s' is large enough. Thus, the adversary gets no information about the challenge bit anymore.

4.3. Non-Interactive Zero-Knowledge Proofs of Knowledge

To instantiate the reputation system, we use the NIZK from [LNP22a]. The relation this NIZK can prove is the following.

Definition 4.6. *Let $q > 0$, $\mathcal{R}$ a ring, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Let $\phi, \phi_{eval}, d, e, v_d, v_e, m_1, \ell, k_{bin} > 0$. Let $\psi : \mathcal{R} \to \mathcal{R}, x \mapsto x(X^{-1})$ be an automorphism. Let*

- $f_i : \mathcal{R}_q^{2(m_1+\ell)} \to \mathcal{R}_q$ *be a quadratic function for* $i \in [\phi]$,
- $F_i : \mathcal{R}_q^{2(m_1+\ell)} \to \mathcal{R}_q$ *be an evaluation function for* $i \in [\phi_{eval}]$,
- $\mathbf{D}_i \in \mathcal{R}_q^{k_i \times 2(m_1+\ell)}, \mathbf{u}_i \in \mathcal{R}_q^{k_i}$ *for* $i \in [v_d]$,
- $\mathbf{E}_i \in \mathcal{R}_q^{p_i \times 2(m_1+\ell)}, \mathbf{v}_i \in \mathcal{R}_q^{p_i}$ *for* $i \in [v_e]$,
- $(\beta_i^{(d)})_{i \in [v_d]}, (\beta_i^{(e)})_{i \in [v_e]}$ *be some bounds,*
- $\mathbf{E}_{bin} \in \mathcal{R}_q^{k_{bin} \times 2(m_1+\ell)}$ *and* $\mathbf{v}_{bin} \in \mathcal{R}_q^{k_{bin}}$.

Call the combination of these parameters pp*. Define the relation* $\mathfrak{R}^R$ *to consist of pairs* $(\mathsf{pp}, \mathbf{s})$ *with* $\mathbf{s} = (\mathbf{s}_1, \psi(\mathbf{s}_1), \mathbf{m}, \psi(\mathbf{m})) \in \mathcal{R}_q^{2m_1} \times \mathcal{R}_q^{2\ell}$*, such that the following conditions hold:*

$$\begin{aligned}
&\forall 1 \leq i \leq \phi, f_i(\mathbf{s}) = 0 \\
&\forall 1 \leq i \leq \phi_{eval}, \tilde{F}_i(\mathbf{s}) = 0 \\
&\forall 1 \leq i \leq v_d, \|\mathbf{D}_i\mathbf{s} - \mathbf{u}_i\|_\infty \leq \beta_i^{(d)} \\
&\forall 1 \leq i \leq v_e, \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\| \leq \beta_i^{(e)} \\
&\mathbf{E}_{bin}\mathbf{s} - \mathbf{v}_{bin} \in \{0,1\}^{dk_{bin}}
\end{aligned}$$

Recall that the notation $\tilde{F}_i(\mathbf{s})$ denotes the constant coefficient of polynomial $F_i(\mathbf{s})$.

Lemma 4.7 ([LNP22a])**.** *There exists a NIZK for relation* $\mathfrak{R}^R$ *that is zero-knowledge and simulation-sound.*

While [LNP22a] only claim soundness instead of simulation-soundness, their analysis ([LNP22b, Appendix B], based on [AFK23]) applies verbatim to simulation-soundness. This is because to argue soundness for a proof π for statement x and message m, one considers only random oracle queries of the form $\mathcal{H}(\mathsf{pp}, x, m, \cdots)$. Simulated proofs for $(x', m') \neq (x, m)$, in contrast, are only concerned with random oracle queries of the form $\mathcal{H}(\mathsf{pp}, x', m', \cdots)$. Hence programming the random oracle for $\mathsf{pp}, x', m', \cdots$ does not interfere with the soundness analysis at all. We can effectively imagine that the simulator and the soundness proof use two independent random oracles.

5. Reputation System

The first step to our reputation system is a syntax model. We base our model on [BJK15], but add some changes. In our model, we define four different (types of) parties: the group manager, the opener, an issuer and a user. In contrast to [BJK15], we identify a user by some user public key upk, which he can generate himself and for which he possesses some user secret key usk. He then can join the reputation system by interacting with the group manager, who knows some group manager key pair $(\mathsf{gmsk}, \mathsf{gmpk})$, with which he generates a registration token ρ to give to the user. Note that the joining of new users is dynamic and the number of users is not limited. The user then interacts with the issuer. The latter is identified by some issuer public key ipk, for which he knows some issuer secret isk. The issuer may then give the user some rating token τ enabling the user to rate the issuer. Note that in contrast to [BJK15], the party to be rated is the issuer and not a product of an issuer. The user can then rate the issuer by using his usk, ρ and τ, where the latter was issued by the issuer to be rated, to create a signature for the rating. Anybody can then verify the signature to check that the rating is valid, while not being able to see which user created the signature. Should the user rate the same issuer twice, anybody can use the linking algorithm to detect that two ratings were created by the same user. The last party is the opener, which in contrast to [BJK15] is a separate party from the group manager. The opener knows some opener secret key osk for some opener public key opk. In the case that a user misbehaves, the opener open a signature to break anonymity of the user, i.e. identify the user who created the signature. Note that the group manager and opener generate their secret keys separately, which is why our model offers a stronger security model than [BJK15]. We now give the formal definition of a reputation system.

Definition 5.1. *A reputation system consists of the following algorithms:*

- $\mathsf{Setup}(1^n)$*: The ppt algorithm outputs some public parameters* pp*. We implicitly assume that all algorithms have* pp *as additional input.*

- $\mathsf{KeyGen}_M(1^n)$*: The ppt algorithm outputs a pair of group manager secret and public key* $(\mathsf{gmsk}, \mathsf{gmpk})$.

- $\mathsf{KeyGen}_O(1^n)$*: The ppt algorithm outputs a pair of opening secret and public key* $(\mathsf{osk}, \mathsf{opk})$.

- $\mathsf{KeyGen}_I(1^n)$*: The ppt algorithm outputs a pair of issuer secret and public key* $(\mathsf{isk}, \mathsf{ipk})$.

- $\mathsf{KeyGen}_U(1^n)$*: The ppt algorithm outputs a pair of user secret and public key* $(\mathsf{usk}, \mathsf{upk})$.

- $\mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}), \mathsf{Register}(\mathsf{gmsk}, \mathsf{upk})$*: At the end of their interaction of these interactive ppt algorithms,* Join *outputs a registration token* ρ.

- $\mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}, \mathsf{usk}, \rho), \mathsf{Issue}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$*: At the end of the interaction of these interactive ppt algorithms,* Request *outputs a rating token τ.*

- $\mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}, \rho, \tau, \mathsf{rtng})$*: The ppt algorithm outputs a signature σ.*

- $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$*. The ppt algorithm outputs a bit b.*

- $\mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$*: The ppt algorithm outputs some* upk.

- $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}', \sigma'), (\mathsf{rtng}'', \sigma''))$*: The ppt algorithm outputs a bit b.*

The correctness of a reputation system is defined as follows.

Definition 5.2. *A reputation system is correct if for all security parameters n,*
all $\mathsf{pp} \in [\mathsf{Setup}(1^n)]$,
all $(\mathsf{gmsk}, \mathsf{gmpk}) \in [\mathsf{KeyGen}_M(1^n)]$,
all $(\mathsf{osk}, \mathsf{opk}) \in [\mathsf{KeyGen}_O(1^n)]$,
all $(\mathsf{isk}, \mathsf{ipk}) \in [\mathsf{KeyGen}_I(1^n)]$,
all $(\mathsf{usk}, \mathsf{upk}_i) \in [\mathsf{KeyGen}_U(1^n)]$,
all $\rho \in [\mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}_i) \leftrightarrow \mathsf{Register}(\mathsf{gmsk}, \mathsf{upk})]$,
all $\tau \in [\mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}, \mathsf{usk}_i, \rho_i) \leftrightarrow \mathsf{Issue}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})]$,
all ratings rtng,
all $\sigma \in [\mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}_i, \rho, \tau, \mathsf{rtng})]$,
all ratings rtng′,
all $\sigma' \in [\mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}, \rho, \tau, \mathsf{rtng}')]$ *it holds that*

- $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma_i) = 1$

- $\mathsf{Open}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma_i) = \mathsf{upk}_i$

- $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}, \sigma), (\mathsf{rtng}', \sigma')) = 1$.

5.1. Security Model

Next we define the security model of a reputation system. We consider five different notions called anonymity, non-frameability, traceability, public-linkability and joining security. These notions are inspired [BJK15], except for non-frameability, which replaces strong-exculpability, and joining security, which is new since we split the group manager and opener into two parties.

In our security games, we model corruption differently than [BJK15] and [EKKS18]. Instead of giving the adversary oracles to corrupt parties, we assume that every participant is corrupted, except for the minimal set that is needed so that the security experiment is not trivially solvable. We note that this model of corruption does not change the security level, it simply makes it easier to argue in proofs. Then, since we differentiate between the group manager and issuers, we can corrupt only one of them if needed. More importantly, this allows us model full corruption, meaning the adversary can choose the public keys *freely* for corrupted parties,where in [EKKS18] the adversary also has to output a valid secret key for the public key he outputs. We also assume that the adversary carries a state in between its calls.

Before we define the security experiments, we define some oracles that an adversary $\mathcal{A}$ may have access to.

$\mathsf{Rg}(\mathsf{gmsk}, \mathsf{upk})$**:** Run $\mathcal{A} \leftrightarrow \mathsf{Register}(\mathsf{gmsk}, \mathsf{upk})$. Add upk to $\mathcal{U}$.

$\mathsf{Req}(\mathsf{gmpk}, \mathsf{ipk}, u)$**:** If the input was queried before, output $\perp$. Else, run $\tau_{u,\mathsf{ipk}} \leftarrow \mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}, \mathsf{usk}_u, \rho_u) \leftrightarrow \mathcal{A}$ and store the rating token $\tau_{u,\mathsf{ipk}}$.

$\mathsf{SigO}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, u, \mathsf{rtng})$**:** If $\tau_{u,\mathsf{ipk}}$ is undefined or the input was queried before, output $\perp$. Else, output $\sigma_{u,\mathsf{ipk}} \leftarrow \mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}_u, \tau_{u,\mathsf{ipk}}, \mathsf{rtng})$. Add $(\mathsf{ipk}, \mathsf{rtng}, \sigma_{u,\mathsf{ipk}})$ to $\mathcal{Q}$.

$\mathsf{Iss}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$**:** Add upk to $\mathcal{I}$. Run $\mathcal{A} \leftrightarrow \mathsf{Issue}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$.

Note that in the security games, some of these parameters are fixed and cannot be chosen by the adversary. For the Rg oracle, for example, we fix the gmsk, but leave the upk argument open and thus write $\mathsf{Rg}(\mathsf{gmsk}, \cdot)$ in the $\mathsf{JoinSecurity}$ game.

The first security requirement for users is that they stay anonymous. In the anonymity experiment, we have two honest users that we try to protect. Except for these two users and the opener, we assume that every other party is corrupted, i.e. controlled by the adversary. In contrast to the notion of full-anonymity of group signature we only have selfless anonymity, meaning it is possible for a user to identify his own signatures. Thus, the usks of the honest users should stay hidden to the adversary.

```
$\mathsf{Anon}_{\Pi,\mathcal{A},b}(n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^n)$
$(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_O(1^n)$
$\mathsf{gmpk} \leftarrow \mathcal{A}(\mathsf{opk})$
For $u \in \{0, 1\}$
    $(\mathsf{usk}_u, \mathsf{upk}_u) \leftarrow \mathsf{KeyGen}_U(1^n)$
    $\rho_u \leftarrow \mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}_u) \leftrightarrow \mathcal{A}(\mathsf{upk}_u)$
    If $\rho_u =\perp$, return 0.
$\mathsf{ipk}^* \leftarrow \mathcal{A}()$
$\tau_u \leftarrow \mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}^*, \mathsf{usk}_u, \rho_u) \leftrightarrow \mathcal{A}$ for $u \in \{0, 1\}$
If $\tau_u =\perp$ for any $u \in \{0, 1\}$, return 0.
$\mathsf{rtng} \leftarrow \mathcal{A}^{\mathsf{Req}(\mathsf{gmpk},\cdot,\cdot),\mathsf{SigO}(\mathsf{gmpk},\mathsf{opk},\cdot,\cdot,\cdot),\mathsf{Open}(\mathsf{gmpk},\mathsf{osk},\cdot,\cdot,\cdot)}$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}^*, \mathsf{usk}_b, \rho_b, \tau_b, \mathsf{rtng})$
$b' \leftarrow \mathcal{A}^{\mathsf{Req}(\mathsf{gmpk},\cdot,\cdot),\mathsf{SigO}(\mathsf{gmpk},\mathsf{opk},\cdot,\cdot,\cdot),\mathsf{Open}(\mathsf{gmpk},\mathsf{isk},\cdot,\cdot,\cdot)}(\sigma)$
If there was a query to $\mathsf{Open}$ with $(\mathsf{gmpk}, \mathsf{osk}, \cdot, \cdot, \sigma)$ as argument, return 0.
If there was a query to $\mathsf{SigO}$ with $(\mathsf{gmpk}, \mathsf{opk}, \cdot, \mathsf{ipk}^*, \cdot)$ as argument, return 0.
Return $b'$.
```

Definition 5.3. *A reputation system* Π *is anonymous, if there exists a negligible function, such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{anon}_{\Pi,\mathcal{A}}(n) := |\Pr[\mathsf{Anon}_{\Pi,\mathcal{A},0}(n) = 1] - \Pr[\mathsf{Anon}_{\Pi,\mathcal{A},1}(n) = 1]| \leq \mathsf{negl}(n).$$

Another security requirement for users is non-frameability. This expresses that any adversary can neither create a signature that opens to an honest user nor create a signature that links to one of an honest user, where the latter security requirement was added by [EKKS18]. In the security experiment, we have one user to be protected. In contrast to [EKKS18], here and in all further security games, we require that the keys of the opener are generated honestly. This is due to the fact that we do not include a Judge algorithm as [EKKS18] do.

```
$\mathsf{NFrame}_{\Pi,\mathcal{A}}(n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^n)$
$\mathcal{Q} = \emptyset$
$(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_O(1^n)$
$\mathsf{gmpk} \leftarrow \mathcal{A}(\mathsf{osk})$
$(\mathsf{usk}_0, \mathsf{upk}_0) \leftarrow \mathsf{KeyGen}_U(1^n)$
$\rho_0 \leftarrow \mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}_0) \leftrightarrow \mathcal{A}(\mathsf{upk}_0)$
$(\mathsf{ipk}, \mathsf{rtng}, \sigma) \leftarrow \mathcal{A}^{\mathsf{Req}(\mathsf{gmpk},\cdot,0),\mathsf{SigO}(\mathsf{gmpk},\mathsf{opk},\cdot,0,\cdot)}()$
$\mathsf{upk} \leftarrow \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$
If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 0$, return 0
If $(\mathsf{ipk}, \mathsf{rtng}, \cdot) \in \mathcal{Q}$, return 0
If $\mathsf{upk} = \mathsf{upk}_0$, return 1
If $\exists(\mathsf{ipk}, \mathsf{rtng}', \sigma') \in \mathcal{Q} : \mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}, \sigma), (\mathsf{rtng}', \sigma')) = 1$, return 1
```

Definition 5.4. *A reputation system* Π *has non-frameability, if there exists a negligible function* negl*, such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

An issuers requires traceability from the reputation system, which means that it is not possible to create a signature that does not open to some user or that opens to a user that was not given a rating token by an honest issuer. Here, we create one honest issuer that we want to protect.

```
$\mathsf{Trace}_{\Pi,\mathcal{A}}(n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^n)$
$\mathcal{I} = \emptyset$
$(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_O(1^n)$
$(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KeyGen}_I(1^n)$
$\mathsf{gmpk} \leftarrow \mathcal{A}(\mathsf{osk}, \mathsf{ipk})$
$(\sigma, \mathsf{rtng}) \leftarrow \mathcal{A}^{\mathsf{Iss}(\mathsf{gmpk},\mathsf{isk},\cdot)}()$
If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 0$, return 0
$\mathsf{upk} \leftarrow \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$
If $\mathsf{upk} = \perp \vee \mathsf{upk} \notin \mathcal{I}$, return 1
```

Definition 5.5. *A reputation system* Π *has traceability, if there exists a negligible function* negl*, such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{Trace}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

A security guarantee for the whole system is public-linkability. This requires that the outputs of Open and Link are consistent to each other, meaning it is not possible for an adversary to create two ratings for the same issuer that open to the same user, but do not link.

```
$\mathsf{PLinkable}_{\Pi,\mathcal{A}}(n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^n)$
$(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_O(1^n)$
$(\mathsf{gmpk}, \mathsf{ipk}, (\sigma_j, \mathsf{rtng}_j)_{j \in \{0,1\}}) \leftarrow \mathcal{A}(\mathsf{osk})$
If $\exists j \in \{0,1\} : \mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}_j, \sigma_j) = 0$, return 0.
If $\mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}_0, \sigma_0) \neq \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}_1, \sigma_1)$, return 0.
If $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}_0, \sigma_0), (\mathsf{rtng}_1, \sigma_1)) = 0$, return 1.
```

Definition 5.6. *A reputation system* Π *has public-linkability, if there exists a negligible function* negl*, such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{PLinkable}_{\Pi,\mathcal{A}}(n) = 1] \leq \mathsf{negl}(n).$$

The group manager also has a security requirement. He wants that every user who wants to join the system must register with him and does not circumvent him. Else, issuers can invent non-existent users to rate themselves or their products.

```
$\mathsf{JoinSecurity}_{\Pi,\mathcal{A}}(n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^n)$
$\mathcal{U} = \emptyset$
$(\mathsf{gmsk}, \mathsf{gmpk}) \leftarrow \mathsf{KeyGen}_M(1^n)$
$(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_O(1^n)$
$(\mathsf{ipk}, \mathsf{rtng}, \sigma) \leftarrow \mathcal{A}^{\mathsf{Rg}(\mathsf{gmsk},\cdot)}(\mathsf{gmpk}, \mathsf{osk})$
If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 0$, return 0.
$\mathsf{upk} \leftarrow \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$
If $\mathsf{upk} \notin \mathcal{U}$, output 1.
```

Definition 5.7. *A reputation system* Π *has join-security, if there exists a negligible function* negl*, such that for all ppt adversaries* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{JoinSecurity}\Pi, \mathcal{A}(n) = 1] \leq \mathsf{negl}(n).$$

5.2. Generic Construction

We construct a reputation system from a signature scheme, an encryption scheme, a LIT and a NIZK.

Construction 5.8. Let $\Sigma = (\mathsf{KeyGen}_\Sigma, \mathsf{Sign}_\Sigma, \mathsf{Vrfy}_\Sigma)$ be a signature scheme. Let $\Pi_{\mathsf{Enc}} = (\mathsf{KeyGen}_{\mathsf{Enc}}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Let $\Pi_{\mathsf{LIT}} = (\mathsf{KeyGen}_{\mathsf{LIT}}, \mathsf{Tag}, \mathsf{Vrfy}_{\mathsf{LIT}}, \mathsf{Link}_{\mathsf{LIT}}, f)$ be a LIT scheme. Let Π_{NIZK} be a non-interactive proof system for the relation listed in the "NIZK" expression below.

- $\mathsf{Setup}(1^n)$: Run $\mathsf{pp} \leftarrow \Pi_{\mathrm{NIZK}}.\mathsf{Setup}(1^n)$.
- $\mathsf{KeyGen}_M(1^n)$: Run $(\mathsf{gmsk}, \mathsf{gmpk}) \leftarrow \mathsf{KeyGen}_\Sigma(1^n)$.
- $\mathsf{KeyGen}_O(1^n)$: Run $(\mathsf{sk}_{\mathsf{Enc}}, \mathsf{pk}_{\mathsf{Enc}}) \leftarrow \mathsf{KeyGen}_{\mathsf{Enc}}(1^n)$ and $(\mathsf{sk}'_{\mathsf{Enc}}, \mathsf{pk}'_{\mathsf{Enc}}) \leftarrow \mathsf{KeyGen}_{\mathsf{Enc}}(1^n)$. Set $(\mathsf{osk}, \mathsf{opk}) = (\mathsf{sk}_{\mathsf{Enc}}, (\mathsf{pk}_{\mathsf{Enc}}, \mathsf{pk}'_{\mathsf{Enc}}))$ and forget $\mathsf{sk}'_{\mathsf{Enc}}$.
- $\mathsf{KeyGen}_I(1^n)$: Run $(\mathsf{isk}, \mathsf{ipk}) \leftarrow \mathsf{KeyGen}_\Sigma(1^n)$.
- $\mathsf{KeyGen}_U(1^n)$: Choose $\mathsf{usk} \leftarrow \mathsf{KeyGen}_{\mathsf{LIT}}(1^n)$ and compute $\mathsf{upk} = f(\mathsf{usk})$.
- $\mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}), \mathsf{Register}(\mathsf{gmsk}, \mathsf{upk})$: The group manager signs $\rho \leftarrow \mathsf{Sign}_\Sigma(\mathsf{gmsk}, \mathsf{upk})$ and sends ρ to the user. If $\mathsf{Vrfy}_\Sigma(\mathsf{gmpk}, \mathsf{upk}, \rho)$, the user outputs it.
- $\mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}, \mathsf{usk}, \rho), \mathsf{Issue}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$: The issuer signs $\tau = \mathsf{Sign}_\Sigma(\mathsf{isk}, \mathsf{upk})$ and sends τ to the user. If $\mathsf{Vrfy}_\Sigma(\mathsf{ipk}, \mathsf{upk}, \tau)$, the user outputs it.

- $\mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}, \rho, \tau, \mathsf{rtng})$: Compute $c = \mathsf{Enc}(\mathsf{pk}_{\mathsf{Enc}}, \mathsf{upk}; r)$. Compute $c' = \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, \mathsf{usk}; r')$. Compute $l = \mathsf{Tag}(\mathsf{usk}, \mathsf{ipk}; r_t)$. Output $\sigma = (c, c', l, \pi)$, where

$$\begin{aligned}\pi = \text{NIZK}\{&\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{pk}_{\mathsf{Enc}}, \mathsf{pk}'_{\mathsf{Enc}}, c, c', l;\\ &\mathsf{upk}, \mathsf{usk}, \rho, \tau, r, r' \ ;\mathsf{upk} = f(\mathsf{usk})\wedge\\ &\mathsf{Vrfy}_{\Sigma}(\mathsf{gmpk}, \mathsf{upk}, \rho) = 1\wedge\\ &\mathsf{Vrfy}_{\Sigma}(\mathsf{ipk}, \mathsf{upk}, \tau) = 1\wedge\\ &c = \mathsf{Enc}(\mathsf{pk}_{\mathsf{Enc}}, \mathsf{upk}; r)\wedge\\ &c' = \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, \mathsf{usk}; r')\wedge\\ &\mathsf{Vrfy}_{\mathsf{LIT}}(\mathsf{usk}, \mathsf{ipk}, l) = 1\}(\mathsf{rtng})\end{aligned}$$

- $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$: Verify π for the corresponding statement.

- $\mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$: Verify π for the corresponding statement. If π is valid, output $\mathsf{upk} = \mathsf{Dec}(\mathsf{osk}, c)$.

- $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}', \sigma'), (\mathsf{rtng}'', \sigma''))$: Verify π', π'' for the corresponding statements. If π', π'' are valid, output $\mathsf{Link}_{\mathsf{LIT}}(\mathsf{ipk}, l', l'')$.

The correctness of the construction follows directly from the correctness of its building blocks.

5.3. Security of the Generic Construction

The encryption of usk with $\mathsf{pk}'_{\mathsf{Enc}}$ in a rating is not necessary for functionality, but a crucial component for the security proof. This is similar to the Naor-Yung paradigm to get CCA security of an encryption scheme from CPA security. Without the encryption of usk we would have to assume simulation-extractability – that it is hard for an adversary to create a valid proof from which an extractor cannot extract, even if the adversary sees simulated proofs for possibly wrong statements not in the language – instead of simulation soundness from the NIZK. This is a significantly stronger assumption on the proof system, so we choose to encrypt the usk and to require simulation-soundness.

Theorem 5.9. *If Π_{Enc} is CPA secure (Definition 2.10), the LIT has indistinguishable tags (Definition 3.2) and Π_{NIZK} has zero-knowledgeness and simulation-soundness (Definitions 2.15 and 2.17), the reputation system is anonymous (Definition 5.3).*

Proof. We prove this by a series of games. An overview can be found in Table 1.

Define $\epsilon_{\mathcal{D},a,b}(n)$ to be the advantage of some ppt $\mathcal{D}$ distinguishing $\mathsf{Game}_a(n)$ from $\mathsf{Game}_b(n)$. Let Game_0 be the Anon_0 game. Define Game_1 to be the same game as Game_0, except that the challenger uses the simulator $\mathcal{S}$ of Π_{NIZK} (Definition 2.15) to generate all proofs, including the challenge. We immediately see that an adversary cannot distinguish between these games, as the difference of the distribution of the proofs is negligible due

	π	Challenge	Query	Tag	Opening
Game_0	$\mathcal{P}$	$c \equiv \mathsf{upk}_0$ $c' \equiv \mathsf{usk}_0$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_0	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_1	$\mathcal{S}$	$c \equiv \mathsf{upk}_0$ $c' \equiv \mathsf{usk}_0$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_0	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_2	$\mathcal{S}$	$c \equiv \mathsf{upk}_0$ $c' \equiv 1^{\lvert\mathsf{usk}_0\rvert}$	$c \equiv \mathsf{upk}_u$ $c' \equiv 1^{\lvert\mathsf{usk}_u\rvert}$	usk_0	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_3	$\mathcal{S}$	$c \equiv \mathsf{upk}_0$ $c' \equiv 1^{\lvert\mathsf{usk}_0\rvert}$	$c \equiv \mathsf{upk}_u$ $c' \equiv 1^{\lvert\mathsf{usk}_u\rvert}$	usk_1	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_4	$\mathcal{S}$	$c \equiv \mathsf{upk}_0$ $c' \equiv \mathsf{usk}_1$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_1	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_5	$\mathcal{S}$	$c \equiv \mathsf{upk}_0$ $c' \equiv \mathsf{usk}_1$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_1	$f(\mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c'))$
Game_6	$\mathcal{S}$	$c \equiv \mathsf{upk}_1$ $c' \equiv \mathsf{usk}_1$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_1	$f(\mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c'))$
Game_7	$\mathcal{S}$	$c \equiv \mathsf{upk}_1$ $c' \equiv \mathsf{usk}_1$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_1	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$
Game_8	$\mathcal{P}$	$c \equiv \mathsf{upk}_1$ $c' \equiv \mathsf{usk}_1$	$c \equiv \mathsf{upk}_u$ $c' \equiv \mathsf{usk}_u$	usk_1	$\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$

Table 1: An overview of the sequence of games for the anonymity proof. The column π states whether proofs are done honestly ($\mathcal{P}$) or simulated ($\mathcal{S}$). The columns *Challenge* and *Query* state what messages are encrypted in the ciphertexts c, c' during the generation of the challenge or the signature query answer. *Tag* states which secret is used to generate a tag. *Opening* states how opening is done.

to the zero-knowledge property of the proof system. Thus, we have that for all ppt distinguishers $\mathcal{D}$, there exists a ppt $\mathcal{A}_0$ such that

$$\mathsf{Adv}^{ZK}_{\Pi_{\mathrm{NIZK}},\mathcal{A}_0}(n) = \epsilon_{\mathcal{D},0,1}(n).$$

Define Game_2 to be the same game as Game_1 except that c' in the signature queries is generated as $c' \leftarrow \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, 1^{|\mathsf{usk}_u|})$, i.e. we encrypt a string of ones instead of usk_u. Furthermore, c' in the challenge is generated as $c' \leftarrow \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, 1^{|\mathsf{usk}_0|})$, i.e. we encrypt a string of ones instead of usk_0. This is indistinguishable by the CPA security of the encryption scheme (Definition 2.10). By a standard hybrid argument we can construct a ppt $\mathcal{A}$ against the CPA security of Π_{Enc} from a distinguisher $\mathcal{D}$ such that

$$\mathsf{Adv}^{CPA}_{\Pi_{\mathsf{Enc}},\mathcal{A}_1}(n) = \frac{1}{Q+1}\epsilon_{\mathcal{D},1,2}(n).$$

Define Game_3 to be the same game as Game_2 except that tags l in the signature queries and the challenge are computed as $l \leftarrow \mathsf{Tag}(\mathsf{usk}_1, \mathsf{ipk}; r_t)$, i.e. we use usk_1 instead of usk_0. This is indistinguishable by the tag-indistinguishability of Π_{LIT} (Definition 3.2). Let $\mathcal{D}$ be distinguisher distinguishing Game_2 and Game_3. Construct an adversary $\mathcal{A}_2$ against the tag-indistinguishability of the LIT.

- On input $(\mathsf{pk}_0, \mathsf{pk}_1)$ set up the reputation system as in Game_2, except for setting $\mathsf{upk}_0 := \mathsf{pk}_0, \mathsf{upk}_1 := \mathsf{pk}_1$.
- Simulate $\mathcal{D}$.
- Whenever $\mathcal{D}$ asks for a signature, query the oracle for a tag l and use that to create the signature. Do the same for the challenge.
- If $\mathcal{D}$ returns a bit b, return b.

We can easily see that if $\mathcal{A}_2$'s challenger is in experiment $b = 0$, then the view of $\mathcal{D}$ is the same as in Game_2, else the view is the same as in Game_3. Thus, we have the following.

$$\mathsf{Adv}^{LITAnon}_{\Pi_{\mathsf{LIT}},\mathcal{A}_2}(n) = \epsilon_{\mathcal{D},2,3}(n)$$

Define Game_4 to be the same game as Game_3 except that c' in the signature queries is generated as $c' \leftarrow \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, \mathsf{usk}_u; r')$, i.e. we again encrypt usk_u instead of $1^{|\mathsf{usk}_u|}$, and c' in the challenge is generated as $c' \leftarrow \mathsf{Enc}(\mathsf{pk}'_{\mathsf{Enc}}, \mathsf{upk}_1; r')$, i.e. we encrypt usk_1 instead of $1^{|\mathsf{usk}_0|}$. By the CPA security of the encryption scheme we immediately have the following for an adversary $\mathcal{A}_3$ that simulates a distinguisher $\mathcal{D}$ as in Game_3, by a similar argument as above:

$$\mathsf{Adv}^{CPA}_{\Pi_{\mathsf{Enc}},\mathcal{A}_3}(n) = \frac{1}{Q+1}\epsilon_{\mathcal{D},3,4}(n)$$

Define Game_5 to be the same game as Game_4 except that opening is done by remembering $\mathsf{sk}'_{\mathsf{Enc}}$ during key generation, decrypting c' to some usk and outputting $f(\mathsf{usk})$

instead of outputting the decryption of c. An adversary can only distinguish between these games if he can submit an opening query $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$ containing ciphertexts c, c' such that the proof is valid but $\mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c) \neq f(\mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c'))$ and such that σ is not an answer he received from the signature oracle. Call the event that an adversary outputs such a query Fake. However, if an adversary could submit such a query, this would break the simulation-soundness of Π_{NIZK} (Definition 2.17). To show this, from a distinguisher $\mathcal{D}$ between Game_4 and Game_5 we construct an adversary $\mathcal{A}_4$ against the simulation-soundness of Π_{NIZK}.

- On input some $\mathsf{pp}_{\mathrm{NIZK}}$, set up Game_4 while remembering $\mathsf{sk}_{\mathsf{Enc}}, \mathsf{sk}'_{\mathsf{Enc}}$ and setting $\mathsf{pp} = \mathsf{pp}_{\mathrm{NIZK}}$.
- Simulate $\mathcal{D}$. To simulate proofs, $\mathcal{A}$ uses its simulator oracle.
- Whenever $\mathcal{D}$ makes an opening query on $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$, answer as in Game_4. Additionally, if $\sigma = (c, c', l, \pi)$ is not an answer from a previous signing query and $\mathsf{upk} \neq \mathsf{upk}'$, where $\mathsf{upk} \leftarrow \mathsf{Dec}(\mathsf{sk}_{\mathsf{Enc}}, c)$ and $\mathsf{upk}' \leftarrow f(\mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c'))$, stop and output the statement from σ together with π.
- If $\mathcal{D}$ stops, output a faliure symbol $\perp$.

If $\mathcal{A}_4$ finds a query such that $\mathsf{upk} \neq \mathsf{upk}'$ and the σ is not from a signature query, we know that, while π is valid and is not a response from the simulator oracle, the statement is not in the language. Therefore, this σ together with the corresponding statement is a proof that breaks the simulation-soundness. Thus, we have that

$$\mathsf{Adv}^{SS}_{\Pi_{\mathrm{NIZK}}, \mathcal{A}_4}(n) = \Pr[\mathsf{Fake}] \geq \epsilon_{\mathcal{D},4,5}(n).$$

Define Game_6 to be the same game as Game_5 except that c in the challenge σ is generated as $c \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathsf{Enc}}, \mathsf{upk}_1; r)$, i.e. we encrypt upk_1 instead of upk_0. This is again indistinguishable by the CPA security of the encryption scheme, thus for a distinguisher $\mathcal{D}$ and an adversary $\mathcal{A}_5$ constructed similarly to above we have

$$\mathsf{Adv}^{CPA}_{\Pi_{\mathsf{Enc}}, \mathcal{A}_5}(n) = \epsilon_{\mathcal{D},5,6}(n)$$

Define Game_7 to be the same game as Game_6 except that opening is done honestly again, i.e. by decrypting c. Again, from a distinguisher $\mathcal{D}$ we can construct an adversary $\mathcal{A}_6$ against the simulation-soundness of Π_{NIZK} similar to above and we get

$$\mathsf{Adv}^{SS}_{\Pi_{\mathrm{NIZK}}, \mathcal{A}_6}(n) \geq \epsilon_{\mathcal{D},6,7}(n)$$

Define Game_8 to be the same game as Game_7 except that the proofs are generated honestly again, thus we have that Game_8 is the same as Anon_1. This is again indistinguishable due to the zero-knowledge property of Π_{NIZK}. Thus, we have that for all distinguishers $\mathcal{D}$, there exists an $\mathcal{A}_7$ such that

$$\mathsf{Adv}^{ZK}_{\Pi_{\mathrm{NIZK}}, \mathcal{A}_7}(n) = \epsilon_{\mathcal{D},7,8}(n).$$

Therefore, in total for any ppt distinguishers $\mathcal{D}_i$ for $i \in \{0, \dots, 7\}$ we have that

$$\begin{aligned}\mathsf{Adv}^{anon}_{\Pi,\mathcal{A}}(n) \leq &\sum_{i=0}^{7} \epsilon_{\mathcal{D}_i,i,i+1}(n)\\ \leq &\ 2\mathsf{Adv}^{ZK}_{\Pi_{\mathrm{NIZK}},\mathcal{A}_0}(n) + \mathsf{Adv}^{LITAnon}_{\Pi_{\mathsf{LIT}},\mathcal{A}_2}(n)\\ &+ (2Q+3)\mathsf{Adv}^{CPA}_{\Pi_{\mathsf{Enc}},\mathcal{A}_1}(n)\\ &+ 2\mathsf{Adv}^{SS}_{\Pi_{\mathrm{NIZK}},\mathcal{A}_4,\mathcal{S}}(n) = 1]\end{aligned}$$

□

Theorem 5.10. *If* Π_{LIT} *is non-invertible and unforgeable (Definitions 3.4 and 3.5) and* Π_{NIZK} *has zero-knowledgeness and simulation-soundness (Definitions 2.15 and 2.17), the reputation system has non-frameability (Definition 5.4).*

Proof. When an adversary against non-frameability wins, we have that the forgery either opens to an honest user or it opens to a user that was not authorized. From these cases, we construct an adversary that targets either the non-invertability or the unforgeability of Π_{LIT}. We also need to analyze the probability of some failure event, for which we use the simulation-soundness of Π_{NIZK}.

Let $\mathcal{A}$ be an adversary against the non-frameability (Definition 5.4) of the reputation scheme that does at most q queries to the signing oracle. Let Fail be the event that in the non-frameability game the statement of the proof contained in the forgery of $\mathcal{A}$ is wrong, i.e. it is not in the language of the relation. Construct an adversary $\mathcal{B}$ against the non-invertability (Definition 3.5) of Π_{LIT} as follows:

- On input pk, set up the environment for $\mathcal{A}$ as in the non-frameability game, except for setting $\mathsf{upk}_0 = \mathsf{pk}$ and remembering $\mathsf{sk}'_{\mathsf{Enc}}$.
- Simulate $\mathcal{A}$. When it queries the request oracle, use the simulator of Π_{NIZK} (cf. Definition 2.15) to answer the query. If it queries the signature oracle, use the tag oracle to generate a tag, generate c, c' honestly, then use the simulator of Π_{NIZK} to generate the proof.
- Eventually, $\mathcal{A}$ outputs some forgery $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$ with $\sigma = (c, c', l, \pi)$. If we have that $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1$ and $u := \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = \mathsf{upk}_0$, decrypt $\mathsf{usk} \leftarrow \mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c')$.
- Output usk.

We can easily see that the view of $\mathcal{A}$ is perfectly simulated, except for negligible error from simulating the proofs. If $\mathcal{A}$ could distinguish the views, we could immediately construct $\mathcal{C}$ that breaks the zero-knowledgeness of Π_{NIZK}. Then, we know that if $\mathcal{A}$ manages to output a valid signature that opens to upk_0, and Fail does not happen, it holds that $\mathsf{pk} = f(\mathsf{usk})$. Thus, we have the following.

$$\Pr[\mathsf{Invert}_{\Pi_{\mathsf{LIT}},\mathcal{B}} = 1] \geq \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 \wedge u = \mathsf{upk}_0 \wedge \neg\mathsf{Fail}] + \mathsf{Adv}^{ZK}_{\Pi_{\mathrm{NIZK}},\mathcal{C}}(n)$$

We also construct a $\mathcal{C}$ against the unforgeability of Π_{LIT} (Definition 3.4).

- Set up the environment for $\mathcal{A}$ as in the non-frameability game, except for setting f to the function provided by the LIT and setting $\mathsf{upk}_0 = \mathsf{pk}$. Also save $\mathsf{sk}'_{\mathsf{Enc}}$.
- Simulate $\mathcal{A}$. When it queries the request oracle, use Π_{NIZK} simulator to answer the query. If $\mathcal{A}$ queries the signature oracle, use the tag oracle to generate a tag, then use the simulator of Π_{NIZK} to answer the query with the corresponding statement.
- $\mathcal{A}$ outputs some $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$ with $\sigma = (c, c', \pi, l)$. If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1$ and $u := \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) \neq \mathsf{upk}_0$ and $\exists(\mathsf{ipk}, \hat{\mathsf{rtng}}, \hat{\sigma}) \in Q$ with $\hat{\sigma} = (\hat{c}, \hat{c}', \hat{\pi}, \hat{l})$ such that $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}, \sigma), (\hat{\mathsf{rtng}}, \hat{\sigma})) = 1$ and $\mathsf{rtng} \neq \hat{\mathsf{rtng}}$, then decrypt $\mathsf{usk} \leftarrow \mathsf{Dec}(\mathsf{sk}'_{\mathsf{Enc}}, c')$ and output $(\mathsf{usk}, \mathsf{ipk}, l)$.

Again, we can easily see that the view of $\mathcal{A}$ is perfectly simulated. If $\mathcal{A}$ outputs a forgery $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$ such that

$$\begin{aligned}
&\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1\\
\text{and } &u := \mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) \neq \mathsf{upk}_0\\
\text{and } &\exists(\mathsf{ipk}, \hat{\mathsf{rtng}}) \in Q : \mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}, \sigma), (\hat{\mathsf{rtng}}, \hat{\sigma})) = 1\\
\text{and } &\mathsf{Fail} \text{ does not happen,}
\end{aligned}$$

we know that by definition we have $\mathsf{Vrfy}_{\mathsf{LIT}}(\mathsf{usk}, \mathsf{ipk}, l) = 1$ and $\mathsf{Link}_{\mathsf{LIT}}(\mathsf{ipk}, l, \hat{l}) = 1$. Therefore we have the following.

$$\Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 | \neg\mathsf{Fail} \wedge u \neq \mathsf{upk}_0] = \Pr[\mathsf{Forge}^{LIT}_{\Pi_{\mathsf{LIT}},C} = 1]$$

Lastly, we want to analyze the probability $\Pr[\mathsf{Fail}]$. For this, we construct an adversary $\mathcal{D}$ against the simulation-soundness of Π_{NIZK} (Definition 2.17) that works as follows:

- On input crs, set up the environment for $\mathcal{A}$ as in the non-frameability game except for using the provided crs.
- Simulate $\mathcal{A}$. Whenever $\mathcal{A}$ makes an oracle query such that the answer would contain a NIZK, use the simulator oracle to generate the proof.
- $\mathcal{A}$ outputs some forgery $(\mathsf{ipk}, \mathsf{rtng}, \sigma)$. If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1$, return σ and the corresponding statement.

We can easily see that $\mathcal{A}$ is perfectly simulated and that if Fail happens, $\mathcal{D}$ wins.

We can then combine the winning conditions of these three adversaries to bound the non-frameability advantage.

$$\begin{aligned}
\Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}}] &\leq \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} \wedge \neg\mathsf{Fail}] + \Pr[\mathsf{Fail}]\\
&= \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 \wedge \neg\mathsf{Fail} \wedge u = \mathsf{upk}_0]\\
&\quad + \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 \wedge \neg\mathsf{Fail} \wedge u \neq \mathsf{upk}_0] + \Pr[\mathsf{Fail}]\\
&\leq \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 | \neg\mathsf{Fail} \wedge u = \mathsf{upk}_0]\\
&\quad + \Pr[\mathsf{NFrame}_{\Pi,\mathcal{A}} = 1 | \neg\mathsf{Fail} \wedge u \neq \mathsf{upk}_0] + \Pr[\mathsf{Fail}]\\
&= \Pr[\mathsf{Invert}_{\Pi_{\mathsf{LIT}},\mathcal{B}} = 1] + \Pr[\mathsf{Forge}^{LIT}_{\Pi_{\mathsf{LIT}},\mathcal{C}} = 1] + \Pr[\mathsf{SimSound}_{\Pi_{\mathrm{NIZK}},\mathcal{D},\mathcal{S}} = 1]
\end{aligned}$$

□

Theorem 5.11. *If Σ is EUF-CMA (Definition 2.12) and Π_{NIZK} is straight-line extractable (Definition 2.18), then the reputation system is traceable (Definition 5.5).*

Proof. Let $\mathcal{E}_0, \mathcal{E}_1$ be the extractor for Π_{NIZK} (cf. Definition 2.18). Let $\mathcal{A}$ be a ppt adversary against traceability.

First, we define $\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n)$ to work like $\mathsf{Trace}_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n)$, except that the public parameters pp are generated by the extractor, i.e. $(\mathsf{pp}, td) \leftarrow \mathcal{E}_0(1^n)$. From the guarantees of the extractor and (Definition 2.18) a straight-forward reduction, we get that $|\Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1] - \Pr[\mathsf{Trace}_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1]| \leq \mathsf{negl}_0(n)$ for some negligible function negl_0.

We now construct an adversary $\mathcal{B}$ against the unforgeability of Σ. $\mathcal{B}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ runs $\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n)$, except that it sets $\mathsf{ipk} = \mathsf{pk}$ (from its input) and whenever $\mathcal{A}$ makes a $\mathsf{Iss}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$ query, $\mathcal{B}$ answers by querying its own oracle $\mathsf{Sign}(\mathsf{sk}, \mathsf{upk})$ for the signature. Eventually, $\mathcal{A}$ outputs (σ, rtng), where $\sigma = (c, c', l, \pi)$. $\mathcal{B}$ runs $\mathcal{E}_1(td, x, \mathsf{rtng}, \pi)$ (where x is set appropriately to the proven statement) to receive a witness $w = (\mathsf{upk}, \mathsf{usk}, \rho, \tau, r, r')$. $\mathcal{B}$ outputs (upk, τ) as a candidate forgery.

Let $\mathsf{fail}_{\mathcal{E}}$ be the event that $\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1$, but $\mathcal{E}_1$ outputs an invalid witness (i.e. $(x, w) \notin \mathfrak{R}$). With a straight-forward reduction to straight-line extractability, we can show that $\Pr[\mathsf{fail}_{\mathcal{E}}] \leq \mathsf{negl}_1(n)$ for some negligible function negl_1.

If $\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1$ and $\neg\mathsf{fail}_{\mathcal{E}}$, $\mathcal{B}$ outputs a valid forgery. This is because the Trace' winning condition "$\mathsf{upk} \notin \mathcal{I}$" (together with $(x, w) \in \mathfrak{R}$ and correctness of the encryption scheme, cf. Definition 2.9) guarantees that $\mathcal{B}$ has not queried its signing oracle for upk with overwhelming probability. Hence there exists a negligible function negl_2 such that

$$\begin{aligned}
&\mathsf{Adv}^{\mathrm{EUFCMA}}_{\Sigma,\mathcal{A}}(n)\\
\geq& \Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1 \wedge \neg\mathsf{fail}_{\mathcal{E}}] - \mathsf{negl}_2(n)\\
=& \Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1] - \Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1 \wedge \mathsf{fail}_{\mathcal{E}}] - \mathsf{negl}_2(n)\\
\geq& \Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1] - \mathsf{negl}_1(n) - \mathsf{negl}_2(n)
\end{aligned}$$

Because $\mathsf{Adv}^{\mathrm{EUFCMA}}_{\Sigma,\mathcal{A}}(n)$ is negligible (given that Σ is EUF-CMA), $\Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1]$ must also be negligible. It follows that $\Pr[\mathsf{Trace}'_{\Pi_{\mathrm{NIZK}},\mathcal{A}}(n) = 1]$ must be negligible, too. □

Theorem 5.12. *If Σ is EUF-CMA (Definition 2.12) and Π_{NIZK} is straight-line extractable (Definition 2.18), then the reputation system has joining security (Definition 5.7).*

The proof is analogous to the proof of Theorem 5.11.

Theorem 5.13. *If Π_{LIT} is linkable (Definition 3.3) and Π_{NIZK} has soundness, the reputation system is publicly linkable (Definition 5.6).*

Proof. Let $\mathcal{A}$ be an adversary against the public linkability of the reputation system. We construct an adversary $\mathcal{B}$ against the linkability of Π_{LIT} from it. $\mathcal{B}$ works as follows:

- Simulate $\mathsf{PLinkable}_{\Pi,\mathcal{A}}(n)$.
- $\mathcal{A}$ outputs some gmpk and ipk and forgery-rating pairs $(\sigma_j, \mathsf{rtng}_j)_{j\in\{0,1\}}$, where $\sigma_j = (c_j, c'_j, l_j, \pi_j)$.
- If both σ_j are valid signatures in the simulated public-linkability game and do not link, decrypt both c'_j to get $\mathsf{usk}_0, \mathsf{usk}_1$ from them and output $(\mathsf{usk}_0, \mathsf{usk}_1, \mathsf{ipk}, l_0, l_1)$.

If $\mathcal{A}$ outputs $\mathsf{gmpk}, \mathsf{ipk}$ with two forgeries σ_0, σ_1 that are valid for these keys and the opk, due to soundness of Π_{NIZK} we have that $\mathsf{Vrfy}_{\mathsf{LIT}}(\mathsf{usk}_j, \mathsf{ipk}, l_j) = 1$ for $j \in \{0,1\}$. Then, again due to the soundness of Π_{NIZK}, we have that $f(\mathsf{usk}_0) = f(\mathsf{usk}_1)$. Call *Sound* the event that $\mathcal{A}$ outputs such tags or such ciphertexts that the above conditions do not hold. Then, we can construct an adversary $\mathcal{C}$ against the soundness of Π_{NIZK}, by simply outputting the proof that $\mathcal{A}$ outputs. Thus, we know that $\Pr[\mathit{Sound}] \leq \mathsf{Adv}^{\mathit{Snd}}_{\Pi,\mathcal{A}}(n)$. If the σ_j do not link, it follows that $(\mathsf{usk}_0, \mathsf{usk}_1, \mathsf{ipk}, l_0, l_1)$ is a tuple of two valid tags for the same message created with $\mathsf{usk}_0, \mathsf{usk}_1$ respectively, which do not link. Therefore, we have that

$$\Pr[\mathsf{Linkable}^{\mathit{LIT}}_{\Pi_{\mathsf{LIT}},\mathcal{C}}(n) = 1] = \Pr[\mathsf{PLinkable}_{\Pi,\mathcal{A}}(n) = 1] + \mathsf{Adv}^{\mathit{Snd}}_{\Pi,\mathcal{A}}(n).$$

□

5.3.1. The Role of Straight-Line Extraction

For the proof of traceability (Theorem 5.11) and joining security (Theorem 5.12), we require Π_{NIZK} to be straight-line extractable, i.e. the proof system must not rely on rewinding for extraction (which, for example, Fiat-Shamir-based proofs usually do). In our security proofs for Theorems 5.11 and 5.12, the reduction algorithm has access to a signature oracle. Similarly to what was noted in [FN16], this represents an issue for an extractor: when rewinding the reduction algorithm $\mathcal{B}$, the extractor needs to answer $\mathcal{B}$'s signing oracle queries. However, in standard definitions, the extractor does not have access to the signing oracle. Even if we grant access, the extractor querying the signing oracle may actually cause an extracted forgery to become invalid. This happens in case a signature on the forgery message is being requested by $\mathcal{B}$ during rewinding. There are potential ways to circumvent this issue for specific proof systems, but standard definitions of (rewinding-based) soundness are incompatible with signing oracle access in security proofs. Straight-line extraction does not suffer from this issue, as the extractor can be used without rewinding.

One can always implement straight-line extractable proofs by encrypting the witness for some honestly generated publicly known public key and proving, with a *sound* zero-knowledge proof, that the encrypted witness is valid. Note that in our security proofs for Theorems 5.11 and 5.12, the only value we need to extract from the proof is the membership certificate τ or ρ (upk is also used, but can be computed by decrypting c). For this reason, when implementing straight-line extractability, it suffices to additionally encrypt τ and ρ, there is no need to encrypt the *full* witness of the rating NIZK.

Alternatively, one can use a NIZK that is inherently straight-line extractable (e.g., using Fischlin's transform [Fis05a]). In practice, one can arguably even use a standard Fiat-Shamir-based construction, for which one cannot prove straight-line extractability (cf. [BNW17]). However, to the best of our knowledge, there is no attack against Fiat-Shamir in practice that targets schemes using it in place of a straight-line extractable proof.

6. A Reputation System from Module Lattices

To instantiate the generic construction based on module lattices, we use the following constructions for the building blocks:

- The LIT is Construction 3.6.
- The signature scheme is Construction 4.3, which is a variant of the signature from [DM14].
- The encryption scheme is the primal Regev-like Construction 4.5.
- The NIZK is the proof system from [LNP22a], described in Lemma 4.7. Note that this NIZK does not offer straight-line extractability as required by the generic construction from Section 5.2, thus we additionally encrypt ρ, τ as mentioned in Section 5.3.1.

We also use a hash function $H : \{0,1\}^* \to \mathcal{R}_q^{n_T \times m_T}$, which will be modeled as a random oracle. Define $\mathsf{BitD} : \mathbb{Z}_q \to \{0,1\}^{\log q}, x \mapsto \mathbf{y}$ such that $\mathbf{y}$ is the bit representation of x. If the argument of BitD is a vector, we apply it component-wise and we interpret $\mathcal{R}_q$ inputs as $\mathbb{Z}_q$ vectors.

Construction 6.1. Let the parameters be as described in Table 2. Note that when choosing concrete parameters, one has to make sure that the bounds from Table 2 are fulfilled and that the building blocks are secure. We omit the parameters for the NIZK for readability. The construction then looks as follows.

- $\mathsf{Setup}(1^n)$: Choose $\mathbf{A}_T \leftarrow \mathcal{R}_q^{n_T \times m_T}$. Then, choose $\bar{\mathbf{A}}_{SE} \leftarrow \mathcal{R}_q^{n_{\mathsf{Enc}} \times m_{\mathsf{Enc}}}$, $\mathbf{S}_{SE} \leftarrow \chi_e^{(4m_\Sigma+2d)\times n_{\mathsf{Enc}}}$, $\mathbf{E}_{SE} \leftarrow \chi_e^{(4m_\Sigma+2d)\times m_{\mathsf{Enc}}}$ and set $\mathbf{B}_{SE} = \mathbf{S}_{SE}\mathbf{A}_{SE} + \mathbf{E}_{SE}$. Set $\mathbf{A}_{SE} = [\bar{\mathbf{A}}_{SE}^t \mid \mathbf{B}_{SE}^t]^t$. Set $\mathsf{pp} = (\mathbf{A}_T, \mathbf{A}_{SE}, \mathbf{B}_{SE})$.
- $\mathsf{KeyGen}_M(1^n)$: Choose $(\mathbf{a}_M^t, \mathbf{R}_M) \leftarrow \mathsf{GenTrap}(1^n, 1^{m_\Sigma}, q)$. Choose $\mathbf{a}_{M,i}^t \leftarrow \mathcal{R}_q^{1\times m_\Sigma}$ for $i \in \{0,\ldots,d\}$. Choose $\mathbf{d}_M^t \leftarrow \mathcal{R}_q^{1\times m_\mu}$, $u_M \leftarrow \mathcal{R}_q$. Set $\mathsf{gmsk} = \mathbf{R}_M$ and $\mathsf{gmpk} = (\mathbf{a}_M^t, \mathbf{a}_{M,0}^t, \ldots, \mathbf{a}_{M,d}^t, \mathbf{d}_M^t, u_M)$.
- $\mathsf{KeyGen}_O(1^n)$: Choose $\bar{\mathbf{A}}_O \leftarrow \mathcal{R}_q^{n_{\mathsf{Enc}} \times m_{\mathsf{Enc}}}, \mathbf{S}_O \leftarrow \chi_e^{(m_T \log q)\times n_{\mathsf{Enc}}}, \mathbf{E}_O \leftarrow \chi_e^{(m_T \log q)\times m_{\mathsf{Enc}}}$ and set $\mathbf{B}_O = \mathbf{S}_O\mathbf{A}_O + \mathbf{E}_O$. Set $\mathbf{A}_O = [\bar{\mathbf{A}}_O^t \mid \mathbf{B}_O^t]^t$. Choose $\mathbf{D}_O \leftarrow \mathcal{R}_q^{(n_T \log q)\times m_{\mathsf{Enc}}}$. Set $\mathsf{opk} = (\mathbf{A}_O, \mathbf{B}_O, \mathbf{D}_O)$ and $\mathsf{osk} = \mathbf{S}_O$.

variable	description	bound
n_t, m_T	dimensions for Π_{LIT}	
β	bound for Π_{LIT}	$\beta < 2^{-\frac{n}{n_T \cdot m_T} + \frac{n}{2n_T}\log(q) - 3}$
n_Σ, m_Σ	dimensions for Π_Σ	$m_\Sigma = \mathcal{O}(n_\Sigma \log q)$
m_μ	message space dim. of Π_Σ	$m_\mu = m_T \cdot \log q$
d	number of mixing lattices of Π_Σ	
s_Σ	parameter for $\mathsf{PreSample}$	$s_\Sigma = n^{3/2} \cdot \omega(\log n)^{3/2}$, such that $s_\Sigma^2 \geq (\sqrt{nm_\Sigma} + \sqrt{nm_\mu} + t)\sqrt{nm_\mu}$
n_{Enc}	dimension for Π_{Enc}	
m_{Enc}	dimension for Π_{Enc}	$m_{\mathsf{Enc}} \geq \max\{4m_\Sigma + 2d, n_U \log q\}$
s_e	parameter for Gaussian error	$s_e < \sqrt{q/4}/\log_2(n)$
s_r	parameter regularity	$s_r > 2n \cdot q^{p/m_{\mathsf{Enc}} + 2/(nm_{\mathsf{Enc}})}$, where $p = \max\{4m_\Sigma + 2d, n_U \log q\}$

Table 2: Parameters for Construction 6.1.

- $\mathsf{KeyGen}_I(1^n)$: Choose $(\mathbf{a}_I^t, \mathbf{R}) \leftarrow \mathsf{GenTrap}(1^n, 1^{m_\Sigma}, q)$. Choose $\mathbf{a}_{I,i}^t \leftarrow \mathcal{R}_q^{1 \times m_\Sigma}$ for $i \in \{0, \ldots, d\}$. Choose $\mathbf{d}_I^t \leftarrow \mathcal{R}_q^{1 \times m_\mu}$, $u_I \leftarrow \mathcal{R}_q$. Set $\mathsf{ipk} = (\mathbf{a}_I^t, \mathbf{a}_{I,0}^t, \ldots, \mathbf{a}_{I,d}^t, \mathbf{d}_I^t, u_I)$ and $\mathsf{isk} = \mathbf{R}_I$.

- $\mathsf{KeyGen}_U(1^n)$: Sample $\mathbf{t} \leftarrow \chi_e^{n_T}, \mathbf{e} \leftarrow \chi_e^{m_T}$. Set $\mathsf{usk} = (\mathbf{t}, \mathbf{e})$ and $\mathsf{upk}^t = \mathbf{t}^t \mathbf{A}_T + \mathbf{e}^t$.

- $\mathsf{Join}(\mathsf{gmpk}, \mathsf{usk}), \mathsf{Register}(\mathsf{gmsk}, \mathsf{upk})$: The group manager chooses $\kappa \leftarrow \mathcal{T}$ and sets $\mathbf{a}_{M,\kappa}^t = [\mathbf{a}_M^t \mid \mathbf{a}_{M,0}^t + \sum_{i=1}^d \kappa_i \mathbf{a}_{M,i}^t]$. The group manager then computes $\sigma' \leftarrow \mathsf{PreSample}(\mathbf{a}_{M,\kappa}^t, (\mathbf{R}_M^t, \mathbf{0})^t, u_M + \mathbf{d}_M^t \mathbf{m}, s_\Sigma)$ and sends $\rho = (\kappa, \sigma)$ to the user. If $\mathsf{Vrfy}_\Sigma(\mathsf{gmpk}, \mathsf{BitD}(\mathsf{upk}), \rho)$, the user outputs it.

- $\mathsf{Request}(\mathsf{gmpk}, \mathsf{ipk}, \mathsf{usk}, \rho), \mathsf{Issue}(\mathsf{gmpk}, \mathsf{isk}, \mathsf{upk})$: The issuer chooses $\kappa \leftarrow \mathcal{T}$ and sets $\mathbf{a}_{I,\kappa}^t = [\mathbf{a}_I^t \mid \mathbf{a}_{I,0}^t + \sum_{i=1}^d \kappa_i \mathbf{a}_{I,i}^t]$. He then computes $\sigma' \leftarrow \mathsf{PreSample}(\mathbf{a}_{I,\kappa}^t, (\mathbf{R}_I^t, \mathbf{0})^t, u_I + \mathbf{d}_I^t \mathbf{m}, s_\Sigma)$ and sends $\tau = (\kappa, \sigma)$ to the user. If $\mathsf{Vrfy}_\Sigma(\mathsf{ipk}, \mathsf{BitD}(\mathsf{upk}), \tau)$, the user outputs it.

- $\mathsf{Sign}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{usk}, \rho, \tau, \mathsf{rtng})$: Choose $\mathbf{r} \leftarrow \chi_r^{m_{\mathsf{Enc}}}$, $\mathbf{r}' \leftarrow \chi_r^{m_{\mathsf{Enc}}}$, $\mathbf{r}_{SE} \leftarrow \chi_r^{m_{\mathsf{Enc}}}$ and compute $\mathbf{c} = \mathbf{A}_O \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor q/2 \rceil \cdot \mathsf{BitD}(\mathsf{upk}) \end{bmatrix}$ and $\mathbf{c}' = \mathbf{D}_O \mathbf{r}' + \begin{bmatrix} \mathbf{0} \\ \lfloor q/2 \rceil \cdot \mathsf{BitD}(\mathsf{usk}) \end{bmatrix}$. Compute $\mathbf{c}_{SE} = \mathbf{A}_{SE} \mathbf{r}_{SE} + \begin{bmatrix} \mathbf{0} \\ \lfloor q/2 \rceil \cdot \mathsf{BitD}((\rho^t \mid \tau^t)^t) \end{bmatrix}$. Compute $\mathbf{A}_{\mathsf{ipk}} = H(\mathsf{ipk})$, choose $\mathbf{e}_l \leftarrow \chi_e^m$, set $\mathbf{l}^t = \mathbf{t}^t\mathbf{A}_{\mathsf{ipk}} + \mathbf{e}_l^t$ and output $\sigma = (\mathbf{c}, \mathbf{c}', \mathbf{l}, \pi)$, where

$$\begin{aligned}
\pi = \mathrm{NIZK}\{&\mathbf{A}_T, \mathbf{A}_M, \mathbf{B}_M, \mathbf{u}_M,\\
&\mathbf{A}_I, \mathbf{B}_I, \mathbf{u}_I, \mathbf{A}_O, \mathbf{B}_O, \mathbf{D}_O, \mathbf{l};\\
&\mathbf{t}, \mathbf{e}, \rho, \tau, r, r', \mathbf{e}_l \;;\; \mathbf{t}^t\mathbf{A}_T + \mathbf{e}^t = \mathsf{upk}^t\\
&\mathsf{Vrfy}_\Sigma((\mathbf{A}_M, \mathbf{B}_M, \mathbf{u}_M), \mathsf{BitD}(\mathsf{upk}), \rho) = 1\wedge\\
&\mathsf{Vrfy}_\Sigma((\mathbf{A}_I, \mathbf{B}_I, \mathbf{u}_I), \mathsf{BitD}(\mathsf{upk}), \tau) = 1\wedge\\
&\mathbf{c} = \mathsf{Enc}([\mathbf{A}_O^t \mid \mathbf{B}_O^t]^t, \mathsf{BitD}(\mathsf{upk}); \mathbf{r})\wedge\\
&\mathbf{c}' = \mathsf{Enc}([\mathbf{A}_O^t \mid \mathbf{D}_O^t]^t, \mathsf{BitD}(\mathsf{usk}); \mathbf{r}')\wedge\\
&\mathbf{c}_{SE} = \mathsf{Enc}([\mathbf{A}_{SE}^t \mid \mathbf{B}_{SE}^t]^t, \mathsf{BitD}((\rho^t \mid \tau^t)^t); \mathbf{r}_{SE})\wedge\\
&\mathbf{l}^t = \mathbf{t}^t\mathbf{A}_{\mathsf{ipk}} + \mathbf{e}_l^t\wedge\\
&\|\mathbf{t}\|_\infty, \|\mathbf{e}\|_\infty, \|\mathbf{e}_l\|_\infty \le \beta\}(\mathsf{rtng}).
\end{aligned}$$

- $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$: Verify π.
- $\mathsf{Open}(\mathsf{gmpk}, \mathsf{osk}, \mathsf{ipk}, \mathsf{rtng}, \sigma)$: Verify σ. If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1$, output $\mathsf{Dec}(\mathsf{osk}, \mathbf{c})$.
- $\mathsf{Link}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, (\mathsf{rtng}, \sigma), (\mathsf{rtng}', \sigma'))$: If $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}, \sigma) = 1$ and $\mathsf{Vrfy}(\mathsf{gmpk}, \mathsf{opk}, \mathsf{ipk}, \mathsf{rtng}', \sigma') = 1$ and $\|\mathbf{l} - \mathbf{l}'\|_\infty < 2\beta$, output 1.

The last part is to argue why we can use the NIZK from [LNP22a]. For this, we have to argue how the statement to prove fits into their framework. We can see that the statement in our signature has to prove four different kinds of sub-statements: possession of a usk for a secret upk, possession of a secret message-signature pair, correctness of an encryption and the possession of an MLWE secret for a public MLWE sample. In the full version of their paper, [LNP22b] already describe how to realize the last two statements (Chapter 6.2 and 6.3).

Proving the possession of a usk for a secret upk in the framework of [LNP22a] is proving that one knows an MLWE secret for an MLWE sample $(\mathbf{A}, \mathbf{b})$, where the $\mathbf{b}$ is secret, i.e. we need to prove possession of $\mathbf{s}_1 = (\mathbf{t}, \mathbf{e}, \mathsf{BitD}(\mathsf{upk}))$ such that $\mathbf{A}_T^t\mathbf{t} + \mathbf{e} = \mathbf{G}\mathsf{BitD}(\mathsf{upk})$ and $\|\mathbf{t}\|_\infty, \|\mathbf{e}\|_\infty \le \beta$ and $\mathsf{BitD}(\mathsf{upk})$ is a bit-vector. We can rewrite the equation as $[\mathbf{A}_T^t \mid \mathbf{I} \mid -\mathbf{G}]\mathbf{s}_1 = \mathbf{0}$, thus we can instantiate the NIZK as shown in Table 3. Technically, we also need to prove that one knows $\mathsf{upk}, \mathsf{BitD}(\mathsf{upk})$ such that $\mathsf{upk} = \mathbf{G} \cdot \mathsf{BitD}(\mathsf{upk})$ and $\mathsf{BitD}(\mathsf{upk})$ is a bit vector, but this can also be realized.

To prove the possession of a secret message-signature pair, we first rewrite the equation from the signature verification to

$$\left[\mathbf{a}^t \mid \mathbf{a}_0^t + \sum_{i=1}^{d} \kappa_i \mathbf{a}_i^t\right] \sigma = u + \mathbf{d}^t\mathbf{m} \Leftrightarrow \left[\mathbf{a}^t \mid \mathbf{a}_0^t \mid \mathbf{a}_1^t \mid \dots \mid \mathbf{a}_d^t \mid -\mathbf{d}^t\right] \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \kappa_1\sigma_2 \\ \vdots \\ \kappa_d\sigma_2 \\ \mathbf{m} \end{pmatrix} = u,$$

variable	description	instantiation
ϕ	# of equations to prove	1
ϕ_{eval}	# of evaluations with const. coeff. zero	0
υ_e	# of exact norm proofs	0
υ_d	# of non-exact norm proofs	2
k_{bin}	length of the binary vector to prove	$m_U \log q$
$\mathbf{s}_1$	committed message in the Ajtai part	$(\mathbf{t}^t, \mathbf{e}^t, \mathsf{BitD}(\mathsf{upk})^t)^t$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$ (no message)
f_1	equation to prove	$\mathbf{A}'\mathbf{s}_1 = 0$
$\mathbf{D}_1$	public matrix for proving $\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\|_\infty \leq \beta_1^{(d)}$	$\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix}$
$\mathbf{u}_1$	public vector for proving $\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\|_\infty \leq \beta_1^{(d)}$	$\mathbf{0}$
$\beta_1^{(d)}$	upper bound on $\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\|_\infty \leq \beta_1^{(d)}$	β
$\mathbf{D}_2$	public matrix for proving $\|\mathbf{D}_2\mathbf{s} - \mathbf{u}_2\|_\infty \leq \beta_2^{(d)}$	$\begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}$
$\mathbf{u}_2$	public vector for proving $\|\mathbf{D}_2\mathbf{s} - \mathbf{u}_2\|_\infty \leq \beta_2^{(d)}$	$\mathbf{0}$
$\beta_2^{(d)}$	upper bound on $\|\mathbf{D}_2\mathbf{s} - \mathbf{u}_2\|_\infty \leq \beta_2^{(d)}$	β
$\mathbf{E}_{bin}$	matrix for proving binary	$\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}$
$\mathbf{v}_{bin}$	vector for proving binary	0

Table 3: Proving possession of a usk for secret upk. Define $\mathbf{A}' = [\mathbf{A}_T^t \mid \mathbf{I} \mid -\mathbf{G}]$.

where $\sigma = (\sigma_1^t, \sigma_2^t)^t$. Therefore, we have an equation that is quadratic in the secret and can thus be proven in the framework of [LNP22a]. To finish proving possession of a secret message-signature pair we additionally need to prove that σ is short and that $\kappa, \mathbf{m}$ are bit vectors, which is also possible in the framework, thus we can instantiate the proof as shown in Table 4.

Since we now have sub-protocols for each component of the proof in the signature of the reputation system, we can combine them into a proof for the whole statement.

6.1. Improved Efficiency via Stateful ℓ-time Signatures

If one wants to deploy the instantiation of the reputation as presented in Section 6 in practice, the size of the ratings may be prohibitively large. This is partly due to the instantiation of the signature scheme with the signatures of [DM14] and the requirement of straight-line extractability of the NIZK: the signatures have security loss depending on the success probability of an adversary, while to realize the straight-line extractability we have to encrypt two signatures, which is costly. Thus, we have two ways to improve the efficiency of our instantiation of the reputation system. One way is to open the black-box and prove that in the instantiation it suffices to have NIZKs that have the weaker notion of extraction“of knowledge”, where the extractor gets oracle access to the proving adversary, such that the extractor can rewind the adversary. The reason that this works is that by opening the black-box one can circumvent the problem of having to extraction in the presence of oracles (cf. Section 5.3.1).

variable	description	instantiation
ϕ	# of equations to prove	1
ϕ_{eval}	# of evaluations with const. coeff. zero	0
υ_e	# of exact norm proofs	1
υ_d	# of non-exact norm proofs	0
k_{bin}	length of the binary vector to prove	$d + m_\mu$
$\mathbf{s}_1$	committed message in the Ajtai part	$(\kappa, \sigma, \mathbf{m})$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$ (no message)
f_1	equation to prove	$\mathbf{a}_\kappa^t \sigma = u + \mathbf{d}^t \mathbf{m}$
$\mathbf{E}_1$	public matrix for proving $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leq \beta_1^{(e)}$	$\begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}$
$\mathbf{v}_1$	public vector for proving $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leq \beta_1^{(e)}$	$\mathbf{0}$
$\beta_1^{(e)}$	upper bound on $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leq \beta_1^{(e)}$	β
$\mathbf{E}_{bin}$	matrix for proving binary	$\mathsf{diag}(\mathbf{I}, \mathbf{0}, \mathbf{I})$
$\mathbf{v}_{bin}$	vector for proving binary	0

Table 4: Proving possession of a Ducas-Micciancio signature (κ, σ) of a message $\mathbf{m}$.

Another way to improve the efficiency of the instantiation of the reputation system is to introduce statefulness and to only allow a limited number of ℓ users. One can change the model of the reputation system such that the group manager and issuers are stateful, i.e. the Join and Issue algorithms get some state as input. The correctness and security model then have to be changed accordingly. Both are is not unreasonable assumptions to make in practice, as group managers have to keep track anyways how many members there are in the system and issuers have to store information about their sales, making both inherently stateful. Furthermore, if ℓ is large enough, e.g. 2^{40} then one can argue that this number of users will not be reached in practice. The security proofs for the reputation system then basically work as before, except that the signatures we use as a building block can be stateful and ℓ-time instead of stateless. However, the straight-line extractability then is a necessary requirement as stateful ℓ-time signatures and extraction via rewinding may not fit together: As explained in Section 5.3.1, extraction during a reduction, where the reduction has access to a signature oracle, can be problematic. If the oracle produces signatures of a stateful signature scheme, the problem worsens. In the first run of the simulated adversary, the reduction will ask the oracle for some signature, while the oracle is in some state st. Assume wlog. that when rewinding the simulated adversary, we rewind the oracle to the state st. Then, the rewound adversary may obviously ask for signatures for different messages than before. However this means the reduction needs to ask the signature oracle for a second signature that was produced from state st, which the signature oracle cannot do, thus we cannot simulate the adversary a second time. Therefore, we need straight-line extraction when using stateful signatures in our construction.

The change to a stateful reputation system allows us to choose different, more efficient signatures than the ones of [DM14]. One example are the signatures described in Appendix C.1, which are a variant of the signatures of [JRLS22] and which are based on MSIS.

Another example are the signatures described in Appendix C.2, which are also a variant of the signatures from [JRLS22] together with a technique from [CEKLL19] and which are based on RSIS, RLWE and NTRU. Depending on whether it is advantageous to be based on MSIS instead of RSIS (i.e. depending on the required degree n of the underlying ring), the latter achieve greater efficiency. This is because the former signatures use a regularity lemma for hiding, while the latter signatures use RLWE to hide, which comes at the cost of also needing the NTRU assumption. Details can be found in the Appendix. [JRLS22] showed that for their signature, one can prove they possess a message-signature pair with the NIZK of [LNP22a]. Since the signatures presented in Appendices C.1 and C.2 are variants of the [JRLS22] signature, it can be shown that proving possession of a message-signature pair of our signatures is also possible with these NIZKs. Thus, we can instantiate the stateful ℓ-time reputation system with these signatures.

6.2. Instantiation with Pairing-Based Cryptography

To instantiate the generic construction based on pairing-based cryptography, we use the following constructions for the building blocks:

- The linking indistinguishable tags are $t = H(\mu)^{\mathsf{usk}}$ with $f(\mathsf{usk}) = g^{\mathsf{usk}}$, where $H : \{0,1\}^* \to \mathbb{G}_1$ is modeled as a random oracle. Two tags t_0, t_1 link if $t_0 = t_1$.
- The signature scheme to sign the user's public key g^{usk} is a simplified version of the structure-preserving signature [Gro15], namely $\sigma = (\tilde{R}, S, T) = (\tilde{g}^r, (y \cdot g^w)^{1/r}, (y^w \cdot M)^{1/r})$ (as in [BEK$^+$21]), where signatures are valid iff they are of that form (can be checked using the pairing).
- The encryption scheme for the user's public key is ElGamal, the encryption scheme for $\mathsf{usk} \in \mathbb{Z}_p$ is bitwise raised Elgamal.[1]
- The NIZK is a simple Schnorr-like protocol made straight-line extractable with Fischlin's transform [Fis05b, KS22].

We leave the details of the instantiation to the reader.

References

[AFK23] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs. In *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part I*, pages 113–142. Springer, 2023.

[1] One can also omit the encryption of usk and instead rely on the simulation-straight-line-extractability of Fischlin's transform to obtain usk in the security proofs that would obtain it via decryption.

[BCOS20] Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. Efficient post-quantum snarks for rsis and rlwe and their applications to privacy. In *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11*, pages 247–267. Springer, 2020.

[BEJ18] Johannes Blömer, Fabian Eidens, and Jakob Juhnke. Practical, anonymous, and publicly linkable universally-composable reputation systems. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 470–490. Springer, 2018.

[BEK$^+$21] Jan Bobolz, Fabian Eidens, Stephan Krenn, Sebastian Ramacher, and Kai Samelin. Issuer-hiding attribute-based credentials. In *CANS*, volume 13099 of *Lecture Notes in Computer Science*, pages 158–178. Springer, 2021.

[BJK15] Johannes Blömer, Jakob Juhnke, and Christina Kolb. Anonymous and publicly linkable reputation systems. In *Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*, pages 478–488. Springer, 2015.

[BJRLW23] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module learning with errors with short distributions. *Journal of Cryptology*, 36(1):1–70, 2023.

[BNW17] David Bernhard, Ngoc Khanh Nguyen, and Bogdan Warinschi. Adaptive proofs have straightline extractors (in the random oracle model). In *Applied Cryptography and Network Security: 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, pages 336–353. Springer, 2017.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography–PKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings 13*, pages 499–517. Springer, 2010.

[BSCR$^+$19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P Ward. Aurora: Transparent succinct arguments for r1cs. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pages 103–128. Springer, 2019.

[BSS10] John Bethencourt, Elaine Shi, and Dawn Song. Signatures of reputation. In Radu Sion, editor, *Financial Cryptography and Data Security, 14th International Conference, FC 2010, Tenerife, Canary Islands, Spain, January 25-28, 2010, Revised Selected Papers*, volume 6052 of *Lecture Notes in Computer Science*, pages 400–407. Springer, 2010.

[CEKLL19] Liqun Chen, Nada El Kassem, Anja Lehmann, and Vadim Lyubashevsky. A framework for efficient lattice-based daa. In *Proceedings of the 1st ACM Workshop on Workshop on Cyber-Security Arms Race*, pages 23–34, 2019.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *Advances in Cryptology-CRYPTO 2006: 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006. Proceedings 26*, pages 78–96. Springer, 2006.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over ntru lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 22–41. Springer, 2014.

[DM14] Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, pages 335–352. Springer, 2014.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.

[DPLS18] Rafaël Del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 574–591, 2018.

[EBEK17] Rachid El Bansarkhani and Ali El Kaafarani. Direct anonymous attestation from lattices. *Cryptology ePrint Archive*, 2017.

[EKKS18] Ali El Kaafarani, Shuichi Katsumata, and Ravital Solomon. Anonymous reputation systems achieving full dynamicity from lattices. In *International*

Conference on Financial Cryptography and Data Security, pages 388–406. Springer, 2018.

[Fis05a] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *Annual International Cryptology Conference*, pages 152–168. Springer, 2005.

[Fis05b] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168. Springer, 2005.

[FN16] Dario Fiore and Anca Nitulescu. On the (in) security of snarks in the presence of oracles. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part I 14*, pages 108–138. Springer, 2016.

[GG21] Stan Gurtler and Ian Goldberg. Sok: Privacy-preserving reputation systems. *Proc. Priv. Enhancing Technol.*, 2021(1):107–127, 2021.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.

[Gro15] Jens Groth. Efficient fully structure-preserving signatures for large messages. In *ASIACRYPT (1)*, volume 9452 of *Lecture Notes in Computer Science*, pages 239–259. Springer, 2015.

[HBB23] Omar Hasan, Lionel Brunie, and Elisa Bertino. Privacy-preserving reputation systems based on blockchain and other cryptographic building blocks: A survey. *ACM Comput. Surv.*, 55(2):32:1–32:37, 2023.

[JRLS22] Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Lattice-based signature with efficient protocols, revisited. *Cryptology ePrint Archive*, 2022.

[KS22] Yashvanth Kondi and Abhi Shelat. Improved straight-line extraction in the random oracle model with applications to signature aggregation. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 279–309. Springer, 2022.

[LLM$^+$16] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pages 373–403. Springer, 2016.

[LM19] Jia Liu and Mark Manulis. prate: Anonymous star rating with rating secrecy. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings*, volume 11464 of *Lecture Notes in Computer Science*, pages 550–570. Springer, 2019.

[LNP22a] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general. In *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II*, pages 71–101. Springer, 2022.

[LNP22b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *Cryptology ePrint Archive*, 2022.

[LNPS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plancon, and Gregor Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27*, pages 218–248. Springer, 2021.

[LNWX17] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Lattice-based group signatures: achieving full dynamicity with ease. In *Applied Cryptography and Network Security: 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings 15*, pages 293–312. Springer, 2017.

[LNWX18] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Constant-size group signatures from lattices. In *Public-Key Cryptography–PKC 2018: 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II 21*, pages 58–88. Springer, 2018.

[LPR13a] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 35–54. Springer, 2013.

[LPR13b] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. Cryptology ePrint Archive, Paper 2013/293, 2013. `https://eprint.iacr.org/2013/293`.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2014.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

A. Uniqueness of (M)LWE Secrets

Definition A.1. *Let $\mathcal{R}$ be a finite ring. For $s \in \mathcal{R} \setminus \{0\}$ set*

$$\mathcal{Z}_s := \{a \in \mathcal{R} \setminus \{0\} : a \cdot s = 0\}$$

and define

$$z_{\max} := \max\{|\mathcal{Z}_s\| : s \in \mathcal{R} \setminus \{0\}\}.$$

Equivalently, $1 + z_{\max}$ is the maximal number of solutions in $\mathcal{R}$ of an equation $x \cdot s = c$, for $s, c \in \mathcal{R}$.

Theorem A.2. *Let $m, k \in \mathbb{N}, D \subseteq \mathcal{R}^k, B \subseteq \mathcal{R}^m$. Then*

$$\Pr\left[\exists(\mathbf{s}, \mathbf{e}) \in D \times B : \mathbf{s} \neq \mathbf{0} \wedge \mathbf{A} \cdot \mathbf{s} = \mathbf{e}; \mathbf{A} \leftarrow \mathcal{R}^{m \times k}\right] \leq \left(\frac{1 + z_{\max}}{|\mathcal{R}|}\right)^m \cdot |D| \cdot |B|.$$

Proof. Fix $(\mathbf{s}, \mathbf{e}), \mathbf{e} = (e_1, \ldots, e_m)$ as in the theorem. By definition of $z_{\max}$

$$\Pr\left[\mathbf{A}_i \cdot \mathbf{s} = e_i; \mathbf{A}_i \leftarrow \mathcal{R}^{1 \times k}\right] \leq \frac{1 + z_{\max}}{|\mathcal{R}|}.$$

Hence for fixed $(\mathbf{s}, \mathbf{e})$

$$\Pr\left[\mathbf{A} \cdot \mathbf{s} = \mathbf{e}; \mathbf{A} \leftarrow \mathcal{R}^{m \times k}\right] \leq \left(\frac{1 + z_{\max}}{|\mathcal{R}|}\right)^m.$$

By the union bound the theorem follows. □

We now restate Lemma 2.4 with a bit more detail in order to prove it.

Lemma A.3 (Short MLWE secrets are unique)**.** *Let $q \neq 2$ be a prime with $q = 3, 5 \mod 8$ (or $q = 1 \mod 2n$), $k > 0$, $n > 16$ be a power of 2, $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$. Let $B_\beta = \{e \in \mathcal{R}_q : \|e\|_\infty \leq \beta\}$. Let $\Delta \geq 0$ such that $2\beta + \Delta < q^{1/4}$. Then there exists some $d < n$ such that*

$$\epsilon(n) := \Pr\left[\begin{array}{l} \exists(\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}') \in (B_\beta^k)^2 \times (B_\beta^m)^2 \\ \textit{with } \mathbf{s} \neq \mathbf{s}' \wedge \|\mathbf{b}\|_\infty \leq \Delta \end{array} : \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{k\times m} \\ \mathbf{b}^t = (\mathbf{s} - \mathbf{s}')^t\mathbf{A} + (\mathbf{e} - \mathbf{e}')^t \end{array}\right] \leq \frac{(4\beta + 2\Delta + 1)^{n(m+k)}}{q^{md}}.$$

Furthermore, there exists an m and a negligible function negl *such that $\epsilon(n) \leq \mathsf{negl}(n)$.*

Proof. Due to the choice of $\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}'$, we know that $\|\mathbf{s}\|_\infty, \|\mathbf{s}'\|_\infty, \|\mathbf{e}\|_\infty, \|\mathbf{e}'\|_\infty \leq \beta$ with overwhelming probability. We define $\hat{\mathbf{s}} = \mathbf{s} - \mathbf{s}' \neq \mathbf{0}$ with $\|\hat{\mathbf{s}}\|_\infty \leq 2\beta$ and $\hat{\mathbf{e}} = \mathbf{e} - \mathbf{e}'$ with $\|\hat{\mathbf{e}}\|_\infty \leq 2\beta$. Then, if $\|\mathbf{A}\hat{\mathbf{s}} + \hat{\mathbf{e}}\|_\infty \leq \Delta$ holds, there exists some $\tilde{\mathbf{e}}$ with $\|\tilde{\mathbf{e}}\|_\infty \leq \Delta$ such that $\mathbf{A}\hat{\mathbf{s}} + \hat{\mathbf{e}} - \tilde{\mathbf{e}} = \mathbf{0}$. Thus we can use Theorem A.2 to show that

$$\begin{aligned} \epsilon(n) :=& \Pr[\exists(\mathbf{s}', \mathbf{e}') \in B_\beta^k \times B_\beta^m : \mathbf{s} \neq \mathbf{s}', \|\mathbf{A}(\mathbf{s} - \mathbf{s}') + \mathbf{e} - \mathbf{e}'\|_\infty \leq \Delta; \mathbf{A} \leftarrow \mathcal{R}_q^{m\times k}, \mathbf{e} \leftarrow \chi^m] \\ =& \Pr[\exists(\hat{\mathbf{s}}, \hat{\mathbf{e}} - \tilde{\mathbf{e}}) \in B_{2\beta}^k \times B_{2\beta+\Delta}^m : \hat{\mathbf{s}} \neq \mathbf{0}, \mathbf{A}\hat{\mathbf{s}} + \hat{\mathbf{e}} - \tilde{\mathbf{e}} = \mathbf{0}; \mathbf{A} \leftarrow \mathcal{R}_q^{m\times k}] \\ \leq& \left(\frac{1 + z_{max}}{|\mathcal{R}_q|}\right)^m \left|B_{2\beta}^k\right| \cdot \left|B_{2\beta+\Delta}^m\right| \end{aligned}$$

By the choice of q and n we know the polynomial $X^n + 1$ is irreducible over $\mathbb{Q}[X]$ and splits over $\mathbb{Z}_q[X]$ into factors of equal degree. Let this degree be d and, accordingly, the number of factors is n/d. Then it holds that

$$\mathcal{R}_q \cong \underbrace{\mathbb{F}_{q^d} \times \cdots \times \mathbb{F}_{q^d}}_{n/d},$$

where $\mathbb{F}_{q^d}$ denotes the field with q^d elements. From this one sees that an element s maximizing $|\mathcal{Z}_s|$ is $(1, 0, \ldots, 0)$ with

$$z_{\max} = |\mathcal{Z}_s| = (q^d)^{n/d-1} - 1 = q^{n-d} - 1.$$

This together with the fact that $|\mathcal{R}_q| = q^n$, results in

$$\epsilon(n) \leq \frac{(4\beta + 1)^{nk} \cdot (4\beta + 2\Delta + 1)^{nm}}{q^{md}} \leq \frac{(4\beta + 2\Delta + 1)^{n(m+k)}}{q^{md}}$$

If $n \geq 16, q = 3, 5 \mod 8$, then one can show that $d = n/2$. In this case, the probability above can be made negligibly small in n for q polynomially large in n and $\beta = q^\gamma, \gamma < 1/4$, even with $m = 1$. If $q = 1 \mod 2n$, then $d = 1$. In this case, for q polynomially in n, one has to pick $m > 1$ to make the probability above negligibly small in n. □

One can also show that uniform secrets of the standard LWE problem are unique, if one chooses m correctly depending on n, q, β.

Corollary A.4 (LWE secrets are unique). *Let q be a prime and $\beta > 0$. Set $B := \{\mathbf{e} \in \mathbb{Z}_q^m : \|\mathbf{e}\|_\infty \leq \beta\}$. Then*

$$\Pr\left[\exists(\mathbf{s},\mathbf{e}) \in \mathbb{Z}_q^n \times B^m : \mathbf{s} \neq \mathbf{0} \wedge \mathbf{A}\cdot\mathbf{s} = \mathbf{e}; \mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}\right] \leq \left(\frac{2\beta+1}{q}\right)^m \cdot q^n.$$

B. Normal-Form Module SIS

Lemma B.1. *Let $\mathcal{R}$ be a finite ring. We denote by*

$$\eta := \Pr[a \textit{ not invertible} : a \leftarrow R]$$

the probability that an element chosen uniformly at random from R is not invertible. Then

$$\Pr[\mathbf{A} \textit{ has a right-inverse} : \mathbf{A} \leftarrow \mathcal{R}^{k\times k}] \geq 1 - k\cdot\eta.$$

Proof. We prove the lemma by induction on k. For $k = 1$ the lemma is immediate from the definition of η. For the induction step, assume

$$\mathbf{A} = (a_{ij})_{1\leq i,j\leq k}, \quad \text{where } a_{ij} \leftarrow \mathcal{R} \text{ for all } i,j.$$

By $\mathbf{A}'$ denote the $(k-1)\times(k-1)$ submatrix of $\mathbf{A}$ consisting of the last $k-1$ rows and columns of $\mathbf{A}$. By induction hypothesis applied to $\mathbf{A}'$ with probability at least $1-(k-1)\eta$ there exists a matrix $\mathbf{T}'$ such that

$$\mathbf{A}\cdot\mathbf{T}' = \begin{bmatrix} a_{11} & * & \cdots & * \\ a_{21} & & & \\ \vdots & & \mathbf{I}_{k-1} & \\ a_{k1} & & & \end{bmatrix}.$$

By further column operations, i.e. another matrix $\mathbf{T}''$, we can further modify $\mathbf{A}$ to obtain

$$\mathbf{A}\cdot\mathbf{T}'\cdot\mathbf{T}'' = \begin{bmatrix} a_{11} + \gamma(\mathbf{A}\setminus\{a_{11}\}) & * & \cdots & * \\ 0 & & & \\ \vdots & & \mathbf{I}_{k-1} & \\ 0 & & & \end{bmatrix},$$

where $\gamma(\mathbf{A}\setminus\{a_{11}\})$ is a term that depends on the entries in $\mathbf{A}$ *except* a_{11}. Since a_{11} is chosen uniformly and independently (from entries in $\mathbf{A}\setminus\{a_{11}\}$) at random,

$$\Pr[a_{11} + \gamma(\mathbf{A}\setminus\{a_{11}\}) \text{ is not invertible} : a_{11} \leftarrow \mathcal{R}] \leq \eta.$$

If $a_{11} + \gamma(\mathbf{A} \setminus \{a_{11}\})$ is invertible, then there exists a matrix $\mathbf{T}'''$ with

$$\mathbf{A} \cdot \mathbf{T}' \cdot \mathbf{T}'' \cdot \mathbf{T}''' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \mathbf{I}_{k-1} & \\ 0 & & & \end{bmatrix},$$

i.e. $\mathbf{T} = \mathbf{T}' \cdot \mathbf{T}'' \cdot \mathbf{T}'''$ is a right-inverse for $\mathbf{A}$. Summarizing,

$$\begin{aligned}&\Pr[\mathbf{A} \text{ does not have a right-inverse} : \mathbf{A} \leftarrow \mathcal{R}^{k\times k}] \leq \\ &\qquad \Pr[\mathbf{A}' \text{ does not have a right-inverse} : \mathbf{A}' \leftarrow \mathcal{R}^{(k-1)\times(k-1)}] + \\ &\qquad\qquad \Pr[a_{11} + \gamma(\mathbf{A} \setminus \{a_{11}\}) \text{ is not invertible} : a_{11} \leftarrow \mathcal{R}] \leq k \cdot \eta,\end{aligned}$$

which proves the lemma. □

The same arguments as in the previous proof can be applied to left-inverses and row operations, we obtain

Corollary B.2. *With the assumptions and notation as in the previous lemma,*

$$\Pr[\mathbf{A} \textit{ has a right- and a left-inverse} : \mathbf{A} \leftarrow \mathcal{R}^{k\times k}] \geq 1 - 2k \cdot \eta.$$

Lemma B.3. *Let $\mathcal{R}$ and η be as above. Assume matrix $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2], \mathbf{A}_1 \in \mathcal{R}^{k\times k}, \mathbf{A}_2 \in \mathcal{R}^{k\times(n-k)}$, is chosen uniformly at random from $\mathcal{R}^{k\times n}, n \geq k$. Then with probability at least $1 - 2k \cdot \eta$, there is a matrix $\mathbf{A}_2'$ such that for $\mathbf{A}' = [\mathbf{I}_k \mid \mathbf{A}_2']$*

$$\Lambda^\perp(\mathbf{A}) = \Lambda^\perp(\mathbf{A}').$$

Proof. By the previous corollary, over the choice of $\mathbf{A}$ with probability $1 - 2k\eta$ matrix $\mathbf{A}_1$ has a left- and a right-inverse. As is well-known, if left- and right-inverses exist, then they are identical. Denote this inverse of $\mathbf{A}_1$ by $\mathbf{A}_1^{-1}$ and set

$$\mathbf{A}' = \mathbf{A}_1^{-1} \cdot \mathbf{A} = [\mathbf{A}_1^{-1} \cdot \mathbf{A}_1 \mid \mathbf{A}_1^{-1} \cdot \mathbf{A}_2] = [\mathbf{I}_k \mid \mathbf{A}_1^{-1} \cdot \mathbf{A}_2].$$

We claim that $\Lambda^\perp(\mathbf{A}) = \Lambda^\perp(\mathbf{A}')$. Since $\mathbf{A} \cdot \mathbf{v} = 0$ implies $\mathbf{A}' \cdot \mathbf{v} = 0$, the inclusion $\Lambda^\perp(\mathbf{A}) \subseteq \Lambda^\perp(\mathbf{A}')$ follows. The other inclusion follows analogously by observing that

$$\mathbf{A}_1^{-1} \cdot \mathbf{A}' = \mathbf{A}_1 \cdot \mathbf{A}_1^{-1} \cdot \mathbf{A} = \mathbf{A}.$$

□

Next we apply this result to rings $\mathbb{Z}_q[X]/(X^n + 1)$ for relevant choices of n and q. To do so we use the following lemma.

Lemma B.4. *Let n be a power of 2 and $q \geq 16$ a prime with $q = 3, 5 \mod 8$. For the ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ with $\mathcal{R} := \mathbb{Z}[X]/(X^n + 1)$ we have*

$$\eta := \Pr[a \textit{ not invertible} : a \leftarrow \mathcal{R}_q] \leq \frac{2}{q^{n/2}}.$$

Proof. For n, q as in the lemma, the polynomial $X^n + 1$ is irreducible over $\mathbb{Q}$ and splits into two irreducible polynomials of degree $n/2$ modulo q. Hence

$$\mathcal{R}_q \cong \mathbb{F}_{q^{n/2}} \times \mathbb{F}_{q^{n/2}},$$

where $\mathbb{F}_{q^{n/2}}$ denotes the field with $q^{n/2}$ elements. Therefore non-invertible elements (zero-divisors and 0) in $\mathcal{R}_q$ are of the form $(0, z)$ or $(z, 0)$ for $z \in \mathbb{F}_{q^{n/2}}$. Hence the number of non-invertible elements is $2q^{n/2} - 1$ and the lemma follows. □

C. Stateful Lattice Signatures

To present our construction of a stateful ℓ-time reputation system, we need stateful ℓ-time signatures, as we use them as a building block. We first define the formal model of such a signature.

Definition C.1 (Stateful Signature Scheme). *A stateful ℓ-time signature scheme Σ consists of the following ppt algorithms:*

- $\mathsf{KeyGen}(1^n)$ *outputs secret key and public key pair* $(\mathsf{sk}, \mathsf{pk})$ *and a state st.*
- $\mathsf{Sign}(\mathsf{sk}, m, st)$ *outputs signature σ and state st'.*
- $\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$ *is deterministic and outputs a bit.*

We say that Σ is correct *if for all $n \in \mathbb{N}$, all* $(\mathsf{sk}, \mathsf{pk}, st_1)$ *output by* $\mathsf{KeyGen}(1^n)$*, all messages $m_1, \ldots, m_\ell$, all $1 \leq i \leq \ell$, and all (st_{i+1}, σ_i) output by* $\mathsf{Sign}(\mathsf{sk}, m_i, st_i)$*, we have* $\mathsf{Vrfy}(\mathsf{pk}, m_i, \sigma_i) = 1$*. We additionally require that for all $1 \leq i \leq \ell$ we have that $|st_i| \leq p(n)$ for some polynomial p.*

To define the EUF-CMA security of a stateful scheme in comparison to a standard stateless EUF-CMA definition, we simply define the signature oracle to remember the (updated) state in between its calls.

Definition C.2 (Stateful EUF-CMA). *A stateful ℓ-time signature scheme Σ is* existentially unforgeable under chosen-message attacks (stateful-EUF-CMA) *if for all ppt $\mathcal{A}$ that make at most ℓ oracle queries,*

$$\begin{aligned}\mathsf{Adv}^{\mathrm{sEUFCMA}}_{\Pi,\mathcal{A}}(n) = \Pr[&\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) = 1 \wedge \mathcal{A} \textit{ has not queried } m : \\ &(\mathsf{sk}, \mathsf{pk}, st_1) \leftarrow \mathsf{KeyGen}(1^n), (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{SigO}(\mathsf{sk},\cdot)}(\mathsf{pk})] \leq \mathsf{negl}(n),\end{aligned}$$

where $\mathsf{SigO}(\mathsf{sk}, m_i)$ *is an oracle that computes $(\sigma_i, st_{i+1}) \leftarrow$* $\mathsf{Sign}(\mathsf{sk}, m_i, st_i)$ *on the ith query, and returns σ_i.*

C.1. Stateful Signatures Based on Module SIS

The first construction of a stateful ℓ-time signature scheme is based on Module SIS and works similar to the construction of [JRLS22]. In comparison to their construction, we do not commit to the message before signing it, which allows us to simplify the construction and the security proof.

Construction C.3. Let $q \geq 2$ with $q = 5 \mod 8$ be an odd prime and let $\zeta, m_3 > 0$. Let $m_1 = k \log q + \omega(\log n)$ and $m_2 = k\zeta$. Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Let the message space be $\mathcal{R}_2^{m_3} \backslash \{\mathbf{0}\}$. Let $g = \lceil q^{\frac{1}{\zeta}} \rfloor$ and $\mathbf{g} = [1 \mid g \mid \ldots \mid g^{\zeta-1}]$ and $\mathbf{G} = \mathbf{I}_d \otimes \mathbf{g} \in \mathcal{R}_q^{k \times m_2}$. Let $s = \eta_\epsilon(\mathbb{Z})\sqrt{1+g^2}\sqrt{1+(\sqrt{nm_1}+\sqrt{nm_2}+t)^2} > 2nq^{k/m_1+2/(nm_1)}$ large enough and $\beta = s\sqrt{n(m_1+m_2)}$ such that $s^2 \geq (\sqrt{nm_1}+\sqrt{nm_3}+t)\sqrt{nm_3}$. Let $\beta = s\sqrt{n(m_1+m_2)}$. Let $w > 0$ and $\mathcal{T}_w = \{e \in \mathcal{R}_2 : \|e\| = \sqrt{w}\}$. Assume there is some order on the elements of $\mathcal{T}_w$. Call κ_i the ith element of $\mathcal{T}_w$ in this order.

- $\mathsf{KeyGen}(1^n)$: Choose $\mathbf{A} \leftarrow \mathcal{R}_q^{k \times m_1}$. Choose $\mathbf{R} \leftarrow \mathcal{R}_{\pm 1}^{m_1 \times m_2}$. Choose $\mathbf{D} \leftarrow \mathcal{R}_q^{k \times m_3}$, $\mathbf{u} \leftarrow \mathcal{R}_q^n$. Set $\mathsf{pk} = (\mathbf{A}, \mathbf{B} = \mathbf{AR}, \mathbf{D}, \mathbf{u})$ and $\mathsf{sk} = \mathbf{R}$. Set $st = \kappa_1$.
- $\mathsf{Sign}(\mathsf{sk}, st, \mathbf{m})$: Set $\mathbf{A}_{\kappa_i} = [\mathbf{A} \mid \mathbf{B} + \kappa_i \mathbf{G}]$ and compute $\sigma \leftarrow \mathsf{PreSample}(\mathbf{A}_{\kappa_i}, -\mathbf{R}, \mathbf{u} + \mathbf{Dm}, s)$. Set $st' = \kappa_{i+1}$. Output $((\kappa, \sigma), st')$.
- $\mathsf{Vrfy}(\mathsf{pk}, \mathbf{m}, (\kappa, \sigma))$: If $\mathbf{A}_\kappa \sigma = \mathbf{u} + \mathbf{Dm}$ and $\|\sigma\| \leq \beta$ and $\kappa \in \mathcal{T}_w$, output 1.

Lemma C.4. *For every ppt adversary that makes at most $|\mathcal{T}_w|$ signature queries and wins the stateful-EUF-CMA game with advantage $\gamma(n)$ against Construction C.3, there exists a ppt adversary against $\mathsf{MSIS}_{\mathcal{R}_q,k,m_1,q,\beta'}$, where*

$$\beta' = \sqrt{1+(\sqrt{nm_1}+\sqrt{nm_2}+t)^2} \cdot (\beta + s\sqrt{n(m_1+m_2)}) + (\sqrt{nm_1}+\sqrt{nm_3}+t)\sqrt{nm_3},$$

with advantage $\frac{1}{|\mathcal{T}_w|}\gamma(n) - \mathsf{negl}(n)$.

Proof. Since the construction is similar to the one of [JRLS22], the proof is similar as well. However, our proof differs in some details.

Let $\mathcal{A}$ be an adversary against the stateful-EUF-CMA security of the signature. From this we construct an adversary $\mathcal{B}$ against MSIS as follows:

- On input $\mathbf{A} \in \mathcal{R}_q^{k \times m_1}$, $\mathcal{B}$ chooses $i^* \leftarrow |\mathcal{T}_w|$. It then chooses $\mathbf{R} \leftarrow \mathcal{R}_{\pm 1}^{m_1 \times m_2}$ and $\mathbf{U} \leftarrow D_{\mathcal{R}_{\pm 1}}^{m_1 \times m_3}$ and $\mathbf{e} \leftarrow D_{\mathcal{R},s}^{m_1+m_2}$. It then sets $\mathbf{u} = \mathbf{A}_{\kappa^*}\mathbf{e}$ and $\mathbf{B} = \mathbf{AR} - \kappa_{i^*}\mathbf{G}$ and $\mathbf{D} = \mathbf{AU}$ and $\mathsf{pk} = (\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u})$.
- $\mathcal{B}$ simulates $\mathcal{A}$ on input pk. On the ith signature query with message $\mathbf{m}$, $\mathcal{B}$ does the following:
  - If $i \neq i^*$, answer with $(\kappa_i, \mathsf{Sign}(-\mathbf{R}, \kappa_i, \mathbf{m}))$.
  - If $i = i^*$, answer with $(\kappa_{i^*}, \mathbf{e}' = \mathbf{e} + \begin{bmatrix} \mathbf{Um} \\ \mathbf{0} \end{bmatrix})$.

- $\mathcal{A}$ outputs some forgery $(m^*, \kappa^*, \sigma^*)$. If $\kappa^* \neq \kappa_{i^*}$, abort.
- $\mathcal{B}$ returns $\mathbf{w} = [\mathbf{I} \mid \mathbf{R}]\,(\sigma^* - \mathbf{e}) - \mathbf{U}\mathbf{m}^*$.

First, we want to argue that the view of $\mathcal{A}$ is correct. For that, we see that $\mathbf{A}$ and the oracle answers in the case $i \neq i^*$ have the same distribution as in the original game. For $\mathbf{B}, \mathbf{D}, \mathbf{u}$ and the oracle answer in the case $i = i^*$ we see that they are computed differently. From [BJRLW23, Lemma 2.8] we know that the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{A}\mathbf{U})$ is statistically close to uniform. Due to Corollary 2.8 we know that there exists a transformation of $\mathbf{A}$ to its normal form with probability $1 - 4kq^{n/2}$. Then with Corollary 7.4 of [LPR13a, LPR13b] we know that $(\mathbf{A}, \mathbf{A}\mathbf{e})$ is statistically close to uniform since $s > 2nq^{k/m_1 + 2/(nm_1)}$. Thus, we lastly have to argue about the distribution of the i^*th query answer. We will argue that the distribution of $\mathbf{e}'$ is statistically indistinguishable from the signature in the real game, when both are conditioned on the respective values of $\mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{u}$. Then we know that the joint distribution of pk together with $\mathbf{e}'$ is statistically indistinguishable. Let $\mathbf{z}$ be some solution to $\mathbf{A}_{\kappa_{i^*}}\mathbf{z} = \mathbf{u}$. Let $\mathbf{c}$ be some solution to $\mathbf{A}_{\kappa_{i^*}}\mathbf{c} = \mathbf{D}\mathbf{m}$. Let $\mathbf{d} = \begin{bmatrix}\mathbf{U}\mathbf{m} \\ \mathbf{0}\end{bmatrix}$. Then, we know that in the original game the distribution of the i^*th signature σ_{i^*} conditioned on pk is $D_{\Lambda^\perp(\mathbf{A}_{\kappa_{i^*}})+\mathbf{z}+\mathbf{d},s} = D_{\Lambda^\perp(\mathbf{A}_{\kappa_{i^*}}),s,-\mathbf{z}-\mathbf{d}} + \mathbf{z} + \mathbf{d}$. If we look at the distribution of $\mathbf{e}$ conditioned on the pk generated by $\mathcal{B}$, we see that its distribution is $D_{\Lambda^\perp(\mathbf{A}_{\kappa_{i^*}})+\mathbf{z},s}$. Thus, the distribution of $\mathbf{e}'$ is $D_{\Lambda^\perp(\mathbf{A}_{\kappa_{i^*}})+\mathbf{z},s} + \mathbf{d} = D_{\Lambda^\perp(\mathbf{A}_{\kappa_{i^*}}),s,-\mathbf{z}} + \mathbf{z} + \mathbf{d}$ and the distributions of σ_{i^*} and $\mathbf{e}'$ only differ in their center. Therefore, from [LSS14, Lemma 4.2] we know that the Rényi difference of the two distributions is smaller than $\exp(\alpha\pi \left\|\mathbf{d}\right\|_2^2 / s^2)$. Then it holds that $\Pr[W_{i^*}]^{\alpha/(\alpha-1)} \leq \exp(\alpha\pi \left\|\mathbf{d}\right\|^2 / s^2) \Pr[W_{\mathbf{e}'}]$ by [JRLS22, Lemma B.1], where W_{i^*} is the event that $\mathcal{A}$ outputs a valid forgery when given σ_{i^*} in the simulation (with changed public keys) and $W_{\mathbf{e}'}$ is the event that $\mathcal{A}$ outputs a valid forgery when given $\mathbf{e}'$. Thus, we know that the probability $\gamma(n)$ that $\mathcal{A}$ outputs a valid forgery in the stateful-EUF-CMA game is smaller than $\gamma(n) \leq \left(\exp(\alpha\pi \left\|\mathbf{d}\right\|^2 / s^2) \Pr[W'_{\mathbf{e}}]\right)^{(\alpha-1)/\alpha} + \mathsf{negl}(n)$.

What is left to argue is that $\mathbf{w}$ is a valid MSIS solution if $\mathcal{A}$ outputs a valid forgery. From the following equation we can see that $\mathbf{w}$ is indeed a vector that maps to $\mathbf{0}$ for the MSIS challenge.

$$\begin{aligned}
&\mathbf{A}_{\kappa^*}\sigma^* = \mathbf{u} + \mathbf{D}\mathbf{m}^* \wedge \mathbf{A}_{\kappa^*}\left(\mathbf{e} + \begin{bmatrix}\mathbf{U}\mathbf{m}_{i^*} \\ \mathbf{0}\end{bmatrix}\right) = \mathbf{u} + \mathbf{D}\mathbf{m}_{i^*} \\
\Rightarrow &\mathbf{A}_{\kappa^*}\sigma^* - \mathbf{D}\mathbf{m}^* = \mathbf{A}_{\kappa^*}\left(\mathbf{e} + \begin{bmatrix}\mathbf{U}\mathbf{m}_{i^*} \\ \mathbf{0}\end{bmatrix}\right) - \mathbf{D}\mathbf{m}_{i^*} \\
\Leftrightarrow &[\mathbf{A} \mid \mathbf{A}\mathbf{R}]\sigma^* - \mathbf{A}\mathbf{U}\mathbf{m}^* = [\mathbf{A} \mid \mathbf{A}\mathbf{R}]\left(\mathbf{e} + \begin{bmatrix}\mathbf{U}\mathbf{m}_{i^*} \\ \mathbf{0}\end{bmatrix}\right) - \mathbf{A}\mathbf{U}\mathbf{m}_{i^*} \\
\Leftrightarrow &\mathbf{A} \cdot ([\mathbf{I} \mid \mathbf{R}](\sigma^* - \mathbf{e}) - \mathbf{U}\mathbf{m}^*) = \mathbf{0}
\end{aligned}$$

Now, for $\mathbf{w}$ to be a valid MSIS solution, it also must be non-zero and short. We follow the heuristic of [JRLS22] for the spectral norms of $\mathbf{R}$ and $\mathbf{U}$ and bound them by

$s_1(\mathbf{R}) \leq \sqrt{nm_1}+\sqrt{nm_2}+t$ and $s_1(\mathbf{U}) \leq \sqrt{nm_1}+\sqrt{nm_3}+t$ for some small t. If one wants to use provable bounds, see [JRLS22, Section 6.2] for details. For the norm of $\mathbf{e}$ one can show that using [MP12, Lemma 2.9] that $\|\mathbf{e}\| \leq s\sqrt{n(m_1+m_2)}$ with overwhelming probability. Therefore, we know that

$$\begin{aligned}\|[\mathbf{I} \mid \mathbf{R}](\sigma^* - \mathbf{e}) - \mathbf{U}\mathbf{m}\| \leq & s_1([\mathbf{I} \mid \mathbf{R}])\left(\|\sigma^*\| + \|\mathbf{e}\|\right) + s_1(\mathbf{U})\|\mathbf{m}\| \\ \leq & \sqrt{1 + (\sqrt{nm_1} + \sqrt{nm_2} + t)^2} \cdot \left(\beta + s\sqrt{n(m_1+m_2)}\right) \\ & + (\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3}\end{aligned}$$

To show that $\mathbf{w}$ is non-zero, we rewrite $\mathbf{w} = \mathbf{y} - \mathbf{U}\mathbf{m}$ for some $\mathbf{y}$. Since we restricted the message space to $\mathcal{R}_q^{m_3}\backslash\{\mathbf{0}\}$, we know that there is at least one column of $\mathbf{U}$ that influences $\mathbf{w}$. Therefore, the adversary has to predict at least one column $\mathbf{u}'$ of $\mathbf{U}$ in order to somehow produce $\mathbf{y}, \mathbf{m}$ such that $\mathbf{w} = \mathbf{0}$. The only places where the adversary might get information about $\mathbf{U}$ from are $\mathbf{D} = \mathbf{A}\mathbf{U}$ and $\mathbf{e}' = \mathbf{e} + \begin{pmatrix}\mathbf{U}\mathbf{m} \\ \mathbf{0}\end{pmatrix} = \mathbf{e}+\mathbf{d}$. However, in a hypothetical game where an unbounded adversary $\mathcal{C}$ tries to predict $\mathbf{u}'$ from information obtained in the stateful-EUF-CMA game, we can gamehop the information about $\mathbf{U}$ in $\mathbf{e}'$ away by replacing $\mathbf{e}'$ by a Gaussian sampled vector and then analyze the probability of predicting a column $\mathbf{u}'$ of $\mathbf{U}$ given $\mathbf{D} = \mathbf{A}\mathbf{U}$. Let V denote the view of $\mathcal{C}$ in the stateful-EUF-CMA game without $\mathbf{D}, \mathbf{e}'$.

$$\begin{aligned}&\Pr\left[\mathbf{u}^* = \mathbf{u}' : \mathbf{u}^* \leftarrow \mathcal{C}(V, \mathbf{D}, \mathbf{e}'), \mathbf{e}' = \mathbf{e} + \mathbf{d}\right] \\ \leq & \left(\Pr[\mathbf{u}^* = \mathbf{u}' : \mathbf{u}^* \leftarrow \mathcal{C}(V, \mathbf{D}, \mathbf{e}''), \mathbf{e}'' \leftarrow D_{\mathcal{R}_q,s}] \cdot \exp(\alpha\pi \|\mathbf{d}\|^2 / s^2)\right)^{\frac{\alpha-1}{\alpha}} \cdot \frac{1+\epsilon}{1-\epsilon} \\ \leq & \left(\mathsf{negl}'(n) \cdot \exp(\alpha\pi \|\mathbf{d}\|^2 / s^2)\right)^{\frac{\alpha-1}{\alpha}} \cdot \frac{1+\epsilon}{1-\epsilon} \\ \leq & \mathsf{negl}(n)\end{aligned}$$

Here, the first inequality follows from [LSS14, Lemma 4.2]. Note that we need the other direction than before, since here we go from a shifted Gaussian to a non-shifted Gaussian. The second inequality follows from [DORS08, Lemma 2.2]: Since there is one column in $\mathbf{D}$ influenced by $\mathbf{u}'$, which has $q^{kn} = 2^{kn\log q}$ possible values, and $\mathbf{u}'$ has Shannon entropy $H_\infty(\mathbf{u}') = \log(3^{m_1}) = m_1 \log_2(3)$, we have

$$\begin{aligned}\tilde{H}_\infty(\mathbf{u}' \mid \mathbf{D}) \geq & H_\infty(\mathbf{u}') - kn\log q \\ & = (kn\log q + \omega(\log n))\log_2(3) - kn\log q \\ & = (\log_2(3) - 1)kn\log q + \log_2(3)\omega(\log n).\end{aligned}$$

Therefore, the probability of guessing a column of $\mathbf{U}$ and thus the probability of $\mathbf{w} = \mathbf{0}$ are negligible. Thus, together with the analysis about the view of $\mathcal{A}$, we know that the probability that $\mathcal{B}$ outputs a valid MSIS solution is greater than

$$\begin{aligned}\Pr[\mathcal{B}\text{ wins}] &\geq (\gamma(n) - \mathsf{negl}(n))^{\alpha/(\alpha-1)} \cdot \exp(-\alpha\pi\|\mathbf{d}\|/s^2) - \mathsf{negl}(n) \\ &\geq (\gamma(n) - \mathsf{negl}(n))^{\alpha/(\alpha-1)} \cdot \exp(-\alpha\pi) - \mathsf{negl}(n)\end{aligned}$$

where the last equation follows since $\|\mathbf{d}\| \leq (\sqrt{nm_1} + \sqrt{nm_3} + t)\sqrt{nm_3} < s^2$. □

C.2. Stateful Signatures Based on RSIS, RLWE and NTRU

Our second construction of a stateful ℓ-time signature works similar to Construction C.3 and thus to the construction of [JRLS22], but is instead based on RSIS, RLWE and NTRU. As explained in Section 6.1, this signature achieves better efficiency under the right conditions. We start with defining the NTRU problem.

Definition C.5 (NTRU)**.** *Let $q > 2$ and $s > 0$ and n be a power of two. Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. The NTRU problem $\mathsf{NTRU}_{q,\mathcal{R},s}$ is to distinguish between a uniform $h \leftarrow \mathcal{R}_q$ and $h = gf^{-1}$, where $f, g \leftarrow D_{\mathcal{R},s}$ such that f is invertible.*

We now construct the signature. Note that apart from some parameters and dimensions, the scheme is the same as Construction C.3.

Construction C.6. Let $q \geq 2$ be odd with $q = 5 \mod 8$ and $\zeta, m_3 > 0$. Let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Let the message space be $\mathcal{R}_2^{m_3} \backslash \{\mathbf{0}\}$. Let $g = \lceil q^{\frac{1}{\zeta}} \rfloor$ and $\mathbf{g} = [1 \mid g \mid \ldots \mid g^{\zeta-1}]$. Let $s = \eta_\epsilon(\mathbb{Z})\sqrt{1+g^2}\sqrt{1 + (\sqrt{2n} + \sqrt{\zeta n} + t)^2} > 2nq^{1/2+n}$ large enough such that $s^2 \geq (\sqrt{2n} + \sqrt{nm_3} + t)\sqrt{nm_3}$. Let $\beta = s\sqrt{n(2+\zeta)}$. Let $w > 0$ and $\mathcal{T}_w = \{e \in \mathcal{R}_2 : \|e\| = \sqrt{w}\}$. Assume there is some order on the elements of $\mathcal{T}_w$. Call κ_i the ith element of $\mathcal{T}_w$ in this order.

- $\mathsf{KeyGen}(1^n)$: Choose $a' \leftarrow \mathcal{R}_q$. Choose $\mathbf{R} \leftarrow \mathcal{R}_{\pm 1}^{2\times\zeta}$. Choose $\mathbf{d} \leftarrow \mathcal{R}_q^{1\times m_3}$, $u \leftarrow \mathcal{R}_q$. Set $\mathbf{a} = [1 \mid a'] \in \mathcal{R}_q^{1\times 2}$. Set $\mathsf{pk} = (\mathbf{a}, \mathbf{b} = \mathbf{aR}, \mathbf{d}, u)$ and $\mathsf{sk} = \mathbf{R}$. Set $st = \kappa_1$.
- $\mathsf{Sign}(\mathsf{sk}, st, \mathbf{m})$: Set $\mathbf{a}_{\kappa_i} = [\mathbf{a} \mid \mathbf{b} + \kappa_i \mathbf{g}]$ and compute $\sigma \leftarrow \mathsf{PreSample}(\mathbf{a}_{\kappa_i}, -\mathbf{R}, u + \mathbf{dm}, s)$. Set $st' = \kappa_{i+1}$. Output (κ, σ, st').
- $\mathsf{Vrfy}(\mathsf{pk}, \mathbf{m}, (\kappa, \sigma))$: If $\mathbf{a}_\kappa \sigma = u + \mathbf{dm}$ and $\|\sigma\| \leq \beta$ and $\kappa \in \mathcal{T}_w$, output 1.

Lemma C.7. *For every ppt adversary that makes at most $|\mathcal{T}_w|$ signature queries and wins the stateful-EUF-CMA game with advantage $\gamma(n)$ against Construction C.6, there exists a ppt adversary against $\mathsf{RSIS}_{q,\mathcal{R},2,\beta'}$, where*

$$\beta' = \sqrt{1 + (\sqrt{2n} + \sqrt{\zeta n} + t)^2} \cdot \left(\beta + s\sqrt{n(2+\zeta)}\right) + s\left(\sqrt{2n} + \sqrt{nm_3} + t\right)\sqrt{nm_3},$$

with advantage greater than $\frac{1}{|\mathcal{T}_w|}\exp(-\alpha\pi)\left(\gamma(n) - \mathsf{negl}(n)\right)^{\alpha/(\alpha-1)} - \mathsf{negl}(n)$, if $\mathsf{RLWE}_{q,\mathcal{R},s}$ and $\mathsf{NTRU}_{q,\mathcal{R},s'}$ are hard.

Proof. The proof works similarly to the one in the construction based on Module SIS by first puncturing the public key at a random tag τ^*, generating $\mathbf{d}$ and u differently with secret information, such that the secret information helps with answering the signature query for τ^*. However, similar to the proof in [CEKLL19], we show that puncturing the key is indistinguishable to the adversary not by some regularity lemma, but by the RLWE assumption. To show this, we temporarily lose the $\mathbf{R}$, with which we generate the signature query answers. Instead we temporarily introduce an NTRU trapdoor to generate the answers.

We prove this formally with the following game hops. Let Game_0 be the original stateful-EUF-CMA game. Let $\gamma(n)$ be the probability that $\mathcal{A}$ wins in this game. Let $\hat{\gamma}_i(n)$ be the probability that $\mathcal{A}$ wins in Game_i.

Game_1 is the same game as Game_0, except that a' is instead set to $a' = gf^{-1}$, where $g, f \leftarrow D_{\mathcal{R}_q,s'}$. This is indistinguishable by the $\mathsf{NTRU}_{q,\mathcal{R},s'}$ assumption. Let $\gamma_{NTRU}(n)$ be the advantage of some ppt adversary against $\mathsf{NTRU}_{q,\mathcal{R},s'}$. Then, we have that $|\hat{\gamma}_0(n) - \hat{\gamma}_1(n)| = \gamma_{NTRU}(n)$.

Game_2 is the same game as Game_1, except that the signature query answers are generated with the NTRU trapdoor instead of $\mathbf{R}$. In particular, by knowing f, g we can construct a basis of the lattice defined by a' such that the norm of its orthogonalization is $1.17\sqrt{q}$ [DLP14]. Since $s \geq 1.17\sqrt{q} \cdot \omega(\sqrt{\log n})$, we can use the GPV preimage sampler [GPV08] to generate the signature query answers with a distribution that is statistically close to the distribution of the scheme. Therefore, we have that $|\hat{\gamma}_1(n) - \hat{\gamma}_2(n)| = \mathsf{negl}(n)$

Game_3 is the same game as Game_2, except that $\mathbf{b} \leftarrow \mathcal{R}_q^{\zeta}$. Immediately, this is indistinguishable by the hardness of normal form $\mathsf{RLWE}_{q,\mathcal{R},s}$. Let $\gamma_{RLWE}(n)$ be the advantage of some ppt adversary against $\mathsf{RLWE}_{q,\mathcal{R},s}$ Then, we have $|\hat{\gamma}_2(n) - \hat{\gamma}_3(n)| = \gamma_{RLWE}(n)$.

Game_4 is the same game as Game_3, except that $\mathbf{b} = \mathbf{b}' - \kappa_{i^*}\mathbf{g}$, where $\mathbf{b}' \leftarrow \mathcal{R}_q^{\zeta}$ and $i^* \leftarrow |\mathcal{T}_w|$. This is indistinguishable since adding a constant to a uniform value does not change the distribution. Thus, we have $|\hat{\gamma}_3(n) - \hat{\gamma}_4(n)| = 0$.

Game_5 is the same game as Game_4, except that $\mathbf{b} = \mathbf{b}' - \kappa_{i^*}\mathbf{g}$, where $\mathbf{b}' = \mathbf{aR} \in \mathcal{R}_q^{\zeta}$ and $\mathbf{R} \leftarrow \mathcal{R}_{\pm 1}^{2\times\zeta}$. This is again indistinguishable due to the hardness of normal form $\mathsf{RLWE}_{q,\mathcal{R},s}$ and we have $|\hat{\gamma}_4(n) - \hat{\gamma}_5(n)| = \gamma_{RLWE}(n)$.

Game_6 is the same game as Game_5, except that $u = [\mathbf{a} \mid \mathbf{b}]\mathbf{e}$ and $\mathbf{d} = \mathbf{aR}'$, where $\mathbf{e} \leftarrow D_{\mathcal{R},s}^{2+\zeta}$ and $\mathbf{R}' \leftarrow D_{\mathcal{R},s}^{m_3 \times m_3}$. Since $s > 2nq^{1/2+n}$ we know from Corollary 7.4 of [LPR13a, LPR13b] that this is statistically indistinguishable. Thus, we have that $|\hat{\gamma}_5(n) - \hat{\gamma}_6(n)| = \mathsf{negl}(n)$.

Game_7 is the same game as Game_6, except the i^*th signature query is instead answered with $(\kappa_{i^*}, \mathbf{e}')$, where $\mathbf{e}' = \mathbf{e} + \begin{pmatrix} \mathbf{Rm} \\ \mathbf{0} \end{pmatrix}$. With a similar argument as in the proof for Lemma C.4, we have $\hat{\gamma}_6(n) \leq (\exp(\alpha\pi \left\|\mathbf{d}\right\| / s^2)\hat{\gamma}_7(n))^{(\alpha-1)/\alpha} \leq (\exp(\alpha\pi)\hat{\gamma}_7(n))^{(\alpha-1)/\alpha}$, since $s^2 \geq (\sqrt{2n} + \sqrt{nm_3} + t)\sqrt{nm_3}$.

Game_8 is the same game as Game_7, except that on the ith signature query, if $i \neq i^*$, the answer is generated as in the original stateful-EUF-CMA game. With a similar argument as before, this is statistically indistinguishable. Thus, we have that $|\hat{\gamma}_7(n) - \hat{\gamma}_8(n)| = \mathsf{negl}(n)$.

Game_9 is the same game as Game_8, except that $a' \leftarrow \mathcal{R}_q$ is chosen uniformly random instead. This is again indistinguishable by the $\mathsf{NTRU}_{q,\mathcal{R},s'}$ assumption and we have $|\hat{\gamma}_8(n) - \hat{\gamma}_9(n)| = \gamma_{NTRU}(n)$.

Therefore, we know that we can simulate $\mathcal{A}$ while having a public key punctured at κ_{i^*}. In total, we have the following.

$$\begin{aligned}\gamma(n) =&\hat{\gamma}_0(n) = \hat{\gamma}_0(n) \sum_{i=1}^{6} -\hat{\gamma}_i(n) + \hat{\gamma}_i(n)\\ \leq&\gamma_{NTRU}(n) + 2\gamma_{RLWE}(n) + \mathsf{negl}(n) + \hat{\gamma}_6(n)\\ \leq&\gamma_{NTRU}(n) + 2\gamma_{RLWE}(n) + \mathsf{negl}(n) + (\exp(\alpha\pi)\hat{\gamma}_7(n))^{(\alpha-1)/\alpha}\\ \leq&\gamma_{NTRU}(n) + 2\gamma_{RLWE}(n) + \mathsf{negl}(n) +\\ &\quad(\exp(\alpha\pi)\left(\gamma_{NTRU}(n) + \hat{\gamma}_9(n) + \mathsf{negl}(n)\right))^{(\alpha-1)/\alpha}\end{aligned}$$

Now we can construct an adversary $\mathcal{B}$ against RSIS that simulates $\mathcal{A}$ in Game_9: On input $\hat{\mathbf{a}} = [\hat{a}_1 \mid \hat{a}_2] \in \mathcal{R}_q^2$, it computes $\mathbf{a} = \hat{a}_1^{-1}\hat{\mathbf{a}}$, which is possible with probability $1 - q^{n/2}$ due to Corollary 2.8. It then simulates $\mathcal{A}$ in Game_9 with that $\mathbf{a}$. When $\mathcal{A}$ outputs a forgery $(m^*, \kappa^*, \sigma^*)$, $\mathcal{B}$ outputs $\mathbf{w} = [\mathbf{I}_2 \mid \mathbf{R}](\sigma^* - \mathbf{e}) - \mathbf{R}'\mathbf{m}^*$.

Having defined this, we can show that with the same arguments as in the proof of Lemma 2.8, but with $k = 1$, that if $\mathcal{A}$ outputs a valid forgery and the guess of i^* was correct, $\mathbf{w}$ is an RSIS solution that is indeed valid, short and non-zero for $\mathbf{a}$ with overwhelming probability. Then we know that $\mathbf{w}$ is also a valid, short, non-zero RSIS solution for $\hat{\mathbf{a}}$, since $\hat{\mathbf{a}}\mathbf{w} = \hat{a}_1\mathbf{a}\mathbf{w} = \mathbf{0}$. Thus we have $\Pr[\mathcal{B} \text{ wins}] = \frac{1}{|\mathcal{T}_w|}\hat{\gamma}_9 - \mathsf{negl}(n)$, where the negligible part is influenced by the probability that $\hat{a}_1$ is invertible and $\mathbf{w}$ being non-zero and short.

In total this means $\mathcal{B}$ solves $\mathsf{RSIS}_{\mathcal{R},q,2,\beta'}$ with probability greater than

$$\Pr[\mathcal{B} \text{ wins}] \geq \frac{1}{|\mathcal{T}_w|} \exp(-\alpha\pi) \left(\gamma(n) - \gamma_{NTRU}(n) - 2\gamma_{RLWE}(n) - \mathsf{negl}(n)\right)^{\alpha/(\alpha-1)} - \gamma_{NTRU}(n) - \mathsf{negl}(n)$$

□