

Traceable Signatures from Lattices

Nam Tran^{a,b}, Khoa Nguyen^a, Dongxi Liu^b, Josef Pieprzyk^{b,c}, Willy Susilo^a

^a*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, 2500, NSW, Australia*

^b*CSIRO's Data61, Marsfield, Sydney, 2122, NSW, Australia*

^c*Institute of Computer Science - Polish Academy of Sciences, Poland*

Abstract

Traceable signatures (Kiayas *et al.*, EUROCRYPT'04) is an anonymous digital signature system that extends the tracing power of opening authority in group signatures. There are many known constructions of traceable signatures but all are based on number-theoretical/pairing assumptions. For such reason, they may not be secure in the presence of quantum computers. This work revisits the notion of traceable signatures and presents a lattice-based construction provably secure in the quantum random oracle model (QROM).

Keywords: post-quantum cryptography, digital signatures, group signatures, traceable signatures, lattice-based cryptography, QROM

1. Introduction

In today's world, balancing privacy with accountability has become a fundamental challenge for designing communication systems. Group signatures, introduced by Chaum and van Heyst [1], was an attempt at resolving this problem by allowing users to sign messages anonymously with a group manager who can trace signatures to their original signers. This dual capability makes group signatures particularly valuable in applications requiring privacy protection and accountability mechanisms. Since their introduction, group signatures have evolved drastically with constructions such as [2, 3, 4, 5, 6], offering increasingly practical solutions. However, these schemes rely on number-theoretical assumptions (either RSA-based or Diffie-Hellman in groups equipped with bilinear pairings) and are vulnerable to attacks from quantum computers. This drawback motivated the search for post-quantum candidates, with Gordon *et al.* [7] giving the first lattice-based group signature scheme in 2010. Since then, this has been an active research area that has developed many constructions with security and efficiency gradually approaching that of their classical counterparts.

Lattice-based constructions are the most dominant among proposals of post-quantum group signatures and remain the most advanced regarding security and efficiency. Although the initial proposal from Gordon *et al.* [7] only achieved weak anonymity with signature size linear order in the number of group members, subsequent works by Laguillaumie *et al.* [8] and others [9, 10, 11] made significant improvement. They

simultaneously reduced the signature size to be logarithmic in group size and achieved anonymity in the strongest sense, where signers' secrets are exposed, and the adversary can open signatures of its choice. Current state-of-the-art constructions are from [12, 13], with a signature size of a few hundred kilobytes for a group size of roughly one million.

Alternatives other than lattices have also been explored, including isogeny-based [14], hash-based [15], and code-based constructions [16, 17, 18]. The code-based scheme from [17] achieves full-anonymity and has a logarithmic signature size in the group size. Recently, [19] proposed a zero-knowledge protocol from the VOLE-in-the-Head framework, which substantially improved upon previous code-based protocols. As an application, they reduced signature size in the scheme from [17] to roughly half a megabyte for a group size of around a billion.

Beyond basic user tracing, incorporating advanced functionalities in group signatures is a related topic that has raised much interest. The need for such features stems from specific privacy and security challenges that the existing means of group signatures may not easily deal with. For example, when the group manager expels some individuals from the group due to misbehavior, it should identify and revoke signatures produced from those. A simple method is to recollect and open every signature, but other users' anonymity would be affected. The notion of verifier local revocable (VLR) group signatures [20] deals with such challenge, by offering a membership-revoking mechanism so that signatures remain valid as long as signers are members of the group. There is also the scenario when the group manager is not fully trusted and needs to be responsible for action. Such is addressed by the notion of accountable tracing signatures [21]. Another interesting functionality is message-dependent opening [22], where the group manager can only open signatures on specific messages with the help of another third party (admitter). In the same spirit, many proposals of privacy-preserving signature systems generalize group signatures by enhancing the tracing power of authority while still protecting users' privacy to a certain extent [23, 24, 25]. This research direction also results in a long line of work extending post-quantum group signatures by incorporating sophisticated and expressive tracing features [26, 27, 28, 29, 24, 25].

This work focuses on traceable signatures, proposed by Kiayias, Tsiounis, and Yung [30]. In essence, traceable signatures are group signatures but offer more tracing power to the group manager and users. In addition to standard tracing capability, group manager also has a *user-specific* tracing power that can be widely distributed among the system. Roughly speaking, the group manager possesses information called *tracing trapdoor* unique to each user. When delegated to sub-openers, the tracing trapdoor helps recognize signatures from the corresponding user while not affecting the privacy of others. In this way, user-specific tracing can be performed concurrently and without recollecting all of the previously issued signatures. In addition, traceable signatures also offer a *self-tracing* mechanism, allowing a user to claim authorship of previously issued signatures without revealing any sensitive information that may compromise its anonymity in the past or the future.

Traceable signatures may find application in anonymous systems where simple tracing is not enough to hold users accountable. For example, in financial transaction systems, there could be situations where a user conducts illegal trading activities. In such cases, the system administrator and other users should be able to invalidate the user’s previous and future transactions. A user-specific tracing method is best for this scenario, which is offered by the means of traceable signatures. Furthermore, a self-tracing capability can allow users to argue that they conducted past transactions faithfully, preventing trading assets from any ownership dispute while maintaining privacy for their other activities.

Functionality	Construction	Post-quantum Assumption and Security Model
Verifier local-revocation	[20, 28, 31, 32]	Lattice-based + ROM [28, 32]
Accountable tracing	[21, 29]	Lattice-based + ROM [29]
Message-dependent opening	[22, 26, 33]	Lattice-based + ROM [26, 33]
User-specific tracing and self-tracing	[30, 34, 35]	✗

Table 1: An overview of some advanced tracing functionalities in group signatures with the corresponding proposals. For constructions from post-quantum assumptions, we list the assumption type and security model in the third column. We remark that for group signatures with user-specific tracing and self-tracing (i.e., traceable signatures), all known constructions are from number-theoretic assumptions. To the best of our knowledge, only [36] proposed a lattice-based traceable signature scheme. However, we identify a critical flaw which will be explained later.

Formalization of Traceable Signatures. Kiayias *et al.* [30] defined the syntax and security requirement of traceable signatures, which we refer to as the KTY model. Informally, a traceable signature scheme in a collection of efficient algorithms (Setup, Join, Sign, Verify, Open, Reveal, Trace, Claim, ClaimVerify). While defining a dynamic group signature scheme, these algorithms extend its functionality by formalizing tracing and claiming mechanisms. The group manager executes Reveal to obtain a user-specific tracing trapdoor. Algorithm Trace allows authorized parties to identify signatures from a specific user using their trapdoor. Algorithm Claim enables users to create claims of authorship for their signatures, using their signing keys as secret input. Such claims can be publicly verified by ClaimVerify.

The KTY security model consists of *traceability*, *non-frameability* and *anonymity*. These are similar to the security notions of group signatures but with subtleties due to the presence of tracing and claiming mechanisms. For example, non-frameability requires that a dishonest group manager or opening authority cannot create a valid signature that is traced or opened to an honest user. Moreover, users cannot illegally claim authorship of signatures they did not create, even when in coalition with the group manager and/or the opening authority. On the other hand, anonymity states that signatures reveal nothing about signers as

long as users' secrets and tracing trapdoors are not exposed. This notion is similar to *selfless-anonymity* in dynamic group signatures, put forward by [37].

There exist extensions of the KTY security model, which generalize the above security notions in a way that resembles those of group signatures. Libert and Yung [34] considered a notion of anonymity similar to *CCA-anonymity*, when the adversary can observe claims and open signatures of its choices. The work of Blazy and Pointcheval [35] generalized the KTY model by separating the roles of opening authority and group manager similar to dynamic group signatures [38]. In their model (the BP model), the opening authority can recover tracing trapdoors from users' public information, and users can deny authorship of signatures.

Previous Constructions. Existing traceable signature schemes have predominantly relied on classical cryptographic assumptions. Kiayas *et al.* [30] proposes the first traceable signature schemes based on pairings and RSA assumptions. Subsequently, constructions such as [39, 40, 34, 35] all were based on variants of Diffie-Hellman-like assumptions and/or required pairings. Surprisingly, to our knowledge, there are no known constructions of traceable signatures from post-quantum assumptions, except for [36]. However, there is a flaw that we will discuss shortly after.

The high-level approach to designing tracing/claiming mechanisms is including a *tracing tag* when generating a signature. Typically, this tag is computed by some pseudorandom function (PRF), evaluated on a seed (or PRF's key) known only by the user and group manager. Constructions such as [39, 40, 34, 35] implicitly employed modified variants of Dodis-Yampolskiy pairing-based verifiable random function [41], which is a special case of PRF. In these constructions, the seed (or PRF's key) will serve as the user-specific tracing trapdoor. Once released, anyone can "test" the tag by either re-evaluating the PRF, or verifying specific equalities the tag must satisfy.

To claim ownership, the user generates another *self-tracing* tag from signing secret and includes it in the signature. Signers create a claim by running a non-interactive zero-knowledge (NIZK) proof showing well-formedness of the tag. Such proof can be obtained from Fiat-Shamir heuristics [42] or by using specific proof systems such as Groth-Sahai proof [43].

We stress that [36] proposed a lattice-based traceable signature scheme and proved security in the random oracle model (ROM). However, in the construction in [36], the tracing tag is computed *deterministically* from the user's secret and remains the same in every user's signature. As such, this does not guarantee anonymity.

Our Contributions. This paper proposes a traceable signature scheme from lattice assumptions and is provably secure in the quantum random oracle model (QROM). Hence, our scheme is the first traceable signature with complete post-quantum security.

Our construction implements the generic framework of dynamic group signatures [38], building upon lattice-based cryptographic primitives including signature scheme by Jeudy *et al.* [44] and GPV identity-

based encryption (IBE) scheme [45]. The non-trivial task is realizing user-specific tracing and self-tracing mechanisms via lattice-based tools. To this end, we adapt the revocation technique of the VLR group signature scheme from [28].

In our scheme, we use the ZK knowledge argument by Yang *et al.* [46] to prove the well-formedness of users' signatures. The relations defining signatures are handled with suitable transformations to make them compatible with the ZK framework. To make the argument a non-interactive argument of knowledge (NIZKAoK) secure in QROM, we employ generalized Unruh transformation [47, 48]. Security of the scheme is proven under the formulation of the BP model [35], which is more expressive and general than that of the KTY model [30]. We stress that the proof is not direct and certain details should be treated carefully using non-blackbox techniques. For example, the anonymity proof uses a technique for simulating GPV-IBE ciphertext in QROM [49].

Overview of Our Techniques. We sketch the high-level idea of tracing and claiming in our construction. For user-specific tracing, we turn to a technique employed by the VLR lattice-based group signature scheme in [28]. Recall that a VLR group signature scheme comes with a membership-revoking functionality. Roughly speaking, the group manager sets up a secret list of *revocation tokens* corresponding to users. A user is revoked from the group when the group manager publishes its token, and signatures are valid as long as the signers' tokens are not revealed. In a sense, this implies user-specific tracing, with revocation tokens playing the same role as tracing trapdoors.

In the VLR group signature scheme of [28], signer includes in signature a tag $\mathbf{t}$ in the form of an LWE sample, i.e. $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e} \bmod q$, where $\mathbf{e} \in \mathbb{Z}^m$ is a randomized error term and $\mathbf{x} \in \mathbb{Z}_q^n$ is the revocation token. The matrix $\mathbf{M} \in \mathbb{Z}_q^{m \times n}$ is also included in the signature and freshly chosen by hashing a short randomness. Based on the LWE assumption, the tag $\mathbf{t}$ hides the token $\mathbf{x}$, guaranteeing the anonymity of users that are not revoked.

Looking differently, $\mathbf{t}$ is a "tag" computed by an LWE-based randomized PRF with "key" $\mathbf{x}$ and "input" $(\mathbf{M}, \mathbf{e})$. From this observation, the revocation technique of [28] can be transformed into a user-specific mechanism, mimicking the PRF-based design in previous traceable schemes. Particularly, in our scheme $\mathbf{t}$ acts as the tracing tag, and $\mathbf{x}$ acts as the user's tracing key. For the self-tracing mechanism, we bind tracing key $\mathbf{x}$ to the user's secret via a suitable one-way function. Signer can claim ownership of a signature by generating an NIZK proof proving knowledge of tracing key $\mathbf{x}$ generating $\mathbf{t}$, and knowledge of the secret that is bound to $\mathbf{x}$. For this purpose, we instantiate the one-way function with the family based on the SIS problem [50]. In this way, $\mathbf{t}$ also acts as the self-tracing tag.

Therefore, with the above modification to the revocation technique of [28], we realize user-specific tracing and self-tracing functionalities using lattice-based cryptographic tools. To construct a full-fledged traceable scheme, we follow the high-level design of dynamic group signatures in [38]. The building blocks consist

of an ordinary digital signature scheme for the group manager to certify members when joining the group, a CCA-secure public key encryption scheme for users to encrypt their identities, and a simulation-sound NIZK proof system for proving relations when generating signatures. For efficiency, we instantiate these components with the lattice-based signature scheme in [44] and the GPV-IBE scheme [45], converted to a CCA-secure encryption with the help of CHK transformation [51].

Regarding the NIZK layer, we employ the ZK argument system in [46] for handling various lattice-based relations. In addition, its inverse polynomial soundness error is an advantage compared to Stern-like ZK framework [52] employed by previous VLR schemes [28, 32]. In particular, one needs $\mathcal{O}(\lambda/\log \text{poly}(\lambda))$ parallel repetitions in order to boost the soundness error of the employed ZK argument system - improving a factor $\log \text{poly}(\lambda)$ to that of Stern ZK protocol. Our design choices give a nearly constant signature size if the group size is a priori bounded.

Other Related Work and Further Discussion. In [40], a different model of traceable signatures was proposed. In this model, tracing agents can compute a set of identifying tokens from trapdoors, allowing them to recognize signatures from specific users. Abe *et al.* [53] provided a generic construction of traceable signatures via a primitive called *double-trapdoor anonymous tag system*.

Although VLR group signatures and traceable signatures bear certain similarities between membership revocation and user-specific tracing, there are a few differences separating the two. First, VLR group signatures can only support static groups, while traceable signatures support dynamic groups. Regarding security, VLR group signatures can achieve the standard security notion of *full-anonymity* [31]. In contrast, the result from [54] potentially suggested that selfless-anonymity is the strongest security traceable signatures can obtain.

Additionally, the syntax of VLR group signatures [20] does not require an opening algorithm. However, this can be done with the help of revocation tokens as follows: the group manager checks for which revocation token such that the signature is not valid when verified; if such token is found, the signature opens to the respective user. In traceable signatures, this is equivalent to opening signatures by trying all available tracing trapdoors instead of using the master opening key. In a sense, traceable signatures imply a form of VLR group signatures for dynamic groups.

From a certain point of view, the result of this paper provides a “conversion” of the VLR signature scheme of [28] to a traceable scheme. Nevertheless, our techniques are non-trivial and depend significantly on the mathematical structure of revocation tags and tokens. Generally speaking, it is unclear whether VLR group signatures can be transformed into traceable signatures. Even the generic blueprint of [31] lacks a proper mechanism linking signers to its signatures to enable ownership claiming. Thus, this may suggest that such a conversion should be non-blackbox and carefully crafted from the underlying design components.

Organization. In Section 2, we recall some results from lattice-based cryptography and the definitions of

traceable signatures. In Section 3, we present our lattice-based traceable signature scheme and then give a detailed analysis of its efficiency and security in Section 4.

Acknowledgement. Nam Tran is supported by CSIRO’s Data61 PhD Scholarship program. Willy Susilo is supported by the Australian Research Council Laureate Fellowship (FL230100033).

2. Preliminaries

2.1. Notations

Let $[n]$ be the set of positive integers up to n , i.e. $[n] = \{1, \dots, n\}$. We work with column vectors and denote them by bold, lower-case letters. Matrices are denoted by bold, upper-case letters. The concatenation of two vectors $\mathbf{v}_1, \mathbf{v}_2$ is a column vector and denoted as $(\mathbf{v}_1, \mathbf{v}_2)$.

For a vector $\mathbf{x} \in \mathbb{R}^n$, the ℓ_2 -norm and ℓ_∞ -norm is defined as

$$\|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^n |\mathbf{x}[i]|^2},$$

$$\|\mathbf{x}\|_\infty := \max\{|\mathbf{x}[1]|, \dots, |\mathbf{x}[n]|\}.$$

For $\mathbf{B} = (\mathbf{b}_1 | \dots | \mathbf{b}_m) \in \mathbb{R}^{n \times m}$, we let $\|\mathbf{B}\|_2 = \max_{i \in [m]} \|\mathbf{b}_i\|_2$ and $\|\mathbf{B}\|_\infty = \max_{i \in [m]} \|\mathbf{b}_i\|_\infty$.

We write $x \leftarrow D$ when x is sampled from a probability distribution D over some space S . In the case D is uniform over some set S , we simply write $x \xleftarrow{\$} S$. For a positive integer m , the m -th dimensional distribution D^m over the space S^m is defined as the distribution of $\mathbf{s} = (s_1, \dots, s_m) \in S^m$, where the components s_i ’s are independently, identically distributed following D .

2.2. Lattices

2.3. Hardness Assumptions

We recall the SIS and the LWE problems, which serve as the hardness assumptions for proving security.

Definition 1. Let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $i \in \{2, \infty\}$, parameters $n, m, q > 0$ and norm bound $0 < B < q$. The SIS problem, denoted by $\text{SIS}_{n,m,q,B}^{(i)}$, asks to find $\mathbf{z} \in \mathbb{Z}_q^m$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{0}$ over $\mathbb{Z}_q$ and $0 < \|\mathbf{z}\|_i \leq B$.

Definition 2. Given $n, q > 0$ and an error distribution χ over $\mathbb{Z}$, the LWE distribution with uniform secret $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ is defined as the distribution $\mathcal{A}_{\mathbf{s}, \chi}$ that works as follows:

- Sample $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$;
- Sample $\mathbf{e} \leftarrow \chi$;
- Compute $\mathbf{b} = \mathbf{a}^\top \cdot \mathbf{s} + \mathbf{e} \bmod q$;

- Output $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

The LWE problem with parameters $n, q > 0$ and an error distribution χ over $\mathbb{Z}$, denoted by $\text{LWE}_{n,q,\chi}$, asks a computationally-bounded adversary $\mathcal{A}$ to distinguish between $m = \text{poly}(n)$ samples of $\mathcal{A}_{\mathbf{s},\chi}$ and m samples drawn from uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

We also work with a variant of LWE [55], in which the adversary has access to a quantum random oracle.

Definition 3 ([55]). Let $n, q, \chi, \mathcal{A}_{\mathbf{s},\chi}$ be defined as in Definition 2 and let $\mathcal{H} : \{0, 1\}^a \rightarrow \{0, 1\}^b$ be a function chosen uniformly at random. The $\text{LWE}_{n,q,\chi,\mathcal{H}}$ problem asks to distinguish $m = \text{poly}(n)$ samples chosen according to $\mathcal{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, given oracle access to $\mathcal{H}$.

It is proven in [55] that a quantum adversary against $\text{LWE}_{n,q,\chi,\text{QRO}_{a,b}}$ can be converted to a quantum adversary against $\text{LWE}_{n,q,\chi}$ with an exact advantage and a loss of the time for reduction depending on the number of queries to the quantum random oracle.

2.3.1. Statistical Lemmas

Lemma 4 ([56, Lemma 4]). Let $\beta = \text{poly}(n)$, $q \geq (4\beta + 1)^2$ and $m \geq 3n$. Then, over the randomness of $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, we have

$$\Pr[\exists \mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\} : \|\mathbf{B} \cdot \mathbf{s} \bmod q\|_\infty \leq 2\beta] \leq q^{-n}.$$

Lemma 5 (Leftover Hash Lemma, adapted from [57, Lemma 1]). Let $q \geq 2$ be a prime and $\mathcal{D}$ be a distribution over $\mathbb{Z}_q^m$ with min-entropy k . For any $\varepsilon > 0$ and $k \geq n \log q + 2 \log(1/\varepsilon) + \mathcal{O}(1)$, the statistical distance between the two distributions

$$\left\{ (\mathbf{A}, \mathbf{A} \cdot \mathbf{y} \bmod q) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{y} \leftarrow \mathcal{D} \right\},$$

$$\left\{ (\mathbf{A}, \mathbf{u}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n \right\}$$

is at most ε .

2.4. Discrete Gaussian and Trapdoor Sampling

For a full-rank lattice $\mathcal{L} \subset \mathbb{Z}^n$, a parameter $\sigma > 0$ and a vector $\mathbf{c} \in \mathbb{R}^m$, the discrete Gaussian distribution over $\mathcal{L}$ with width σ and center $\mathbf{c}$, denoted as $\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}}$ is defined such that for all $\mathbf{x} \in \mathcal{L}$ the value $\mathcal{D}_{\mathcal{L},\sigma,\mathbf{c}}(\mathbf{x})$ is proportional to $\rho_{\sigma,\mathbf{c}}(\mathbf{x})$ where $\rho_{\sigma,\mathbf{c}}$ is the Gaussian function:

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\frac{\pi \|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}\right), \forall \mathbf{x} \in \mathbb{R}^n.$$

When the center $\mathbf{c}$ is zero, the discrete Gaussian is denoted by $\mathcal{D}_{\mathcal{L},\sigma}$. When $\mathcal{L} = \mathbb{Z}^n$, we write $\mathcal{D}_\sigma^n$ to denote the discrete Gaussian $\mathcal{D}_{\mathbb{Z}^n,\sigma}$. We stress that $\mathcal{D}_\sigma^n$ is identical to the distribution $(\mathcal{D}_{\mathbb{Z},\sigma})^n$.

Theorem 6 ([58, 59]). *For a full-rank lattice $\mathcal{L} \subset \mathbb{Z}^n$ and $\sigma > 0$, the following hold*

1. $\Pr_{\mathbf{v} \leftarrow \mathcal{D}_{\mathcal{L}, \sigma}}[\|\mathbf{v}\|_2 > \sigma \cdot \sqrt{n}] \leq 2^{-2n};$
2. *A set of $\mathcal{O}(n^2)$ vectors independently sampled from $\mathcal{D}_{\mathcal{L}, \sigma}$ contains n linearly independent vectors except with probability at most $\text{negl}(n)$.*

We also recall the notion of *gadget matrix* in [60], that is a matrix $\mathbf{G}_n \in \mathbb{Z}_q^{n \times m}$ of the form

$$\left(\mathbf{I}_n \otimes \begin{pmatrix} 1 & 2 & \dots & 2^{\lceil \log q \rceil - 1} \end{pmatrix} \mid \mathbf{0}^{m-n \lceil \log q \rceil} \right) \in \mathbb{Z}_q^{n \times m}.$$

The row vector $\mathbf{g} := \begin{pmatrix} 1 & 2 & \dots & 2^{\lceil \log q \rceil - 1} \end{pmatrix}$ is called *gadget vector*. We remark that for any $\mathbf{u} \in \mathbb{Z}_q^n$, there exists a unique vector $\mathbf{v} \in \{0, 1\}^m$ such that $\mathbf{G}_n \cdot \mathbf{v} = \mathbf{u} \bmod q$. We call $\mathbf{v}$ the binary decomposition of $\mathbf{u}$ along $\mathbf{G}_n$ and denote as $\text{bin}(\mathbf{u})$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a $\mathbf{G}$ -trapdoor of $\mathbf{A}$ with an invertible tag matrix $\mathbf{H} \in \mathbb{Z}_q^n$ is a short matrix $\mathbf{R} \in \mathbb{Z}^{(m-n \lceil \log q \rceil) \times n \lceil \log q \rceil}$ such that $\mathbf{A} \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{I}_{n \lceil \log q \rceil} \end{pmatrix} = \mathbf{H} \cdot \mathbf{G}$.

We also make use of the $\mathbf{G}$ -trapdoor sampling algorithm from [60].

Proposition 1 ([60]). *There exists an algorithm SampleD that takes as input a matrix $\mathbf{R} \in \mathbb{Z}^{m \times n \lceil \log q \rceil}$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ and a standard deviation $\sigma \geq \omega(\sqrt{\log n}) \cdot \sqrt{1 + \|\mathbf{R}\|_2^2}$, outputs $\mathbf{v} \in \mathbb{Z}_q^{m+n \lceil \log q \rceil}$. The distribution of $\mathbf{v}$ is at a statistical distance at most $\text{negl}(n)$ to $\mathcal{D}_{\sigma}^{m+n \lceil \log q \rceil}$ conditioned on $(\mathbf{A} | \mathbf{H}\mathbf{G} - \mathbf{A}\mathbf{R}) \mathbf{v} = \mathbf{u} \bmod q$.*

2.5. Traceable Signatures

The syntax of *traceable signatures* (TS) follows the BP model [35]. A TS scheme $\mathcal{TS}$ is a tuple of polynomial-time algorithms (Setup, KeyGen, Join, Sign, Verify, Open, Reveal, Trace, Claim, ClaimVerify) such that:

Setup($1^\lambda, N$) $\rightarrow$ **pp**. It is a probabilistic algorithm that takes a security parameter λ and a group size N as input; and outputs public parameters **pp** which are available to all users. Algorithms defining $\mathcal{TS}$ may implicitly take **pp** as input.

KeyGen(1^λ) $\rightarrow$ (**gpk**, **gsk**, **osk**, **reg**). It is a probabilistic algorithm that takes as input security parameter λ and generates a group public key **gpk**, a secret key **gsk** of a group manager $\mathcal{GM}$ and a secret key **osk** of an opening authority $\mathcal{OA}$. It also initialized a registry table **reg** recording public information of users.

Join($\langle \mathcal{GM}(\mathbf{gsk}), \mathcal{U} \rangle$ (**gpk**)). This is an interactive protocol between $\mathcal{GM}$ (holding **gsk**) and a user $\mathcal{U}$ who wants to join the group. Both parties rely on **gpk** as the common information. At the end of Join:

- user $\mathcal{U}$ obtains $(\text{id}, \text{usk}_{\text{id}}, \text{cert}_{\text{id}})$, where id is a unique, public user identifier, usk_{id} is secret known only by the user and cert_{id} is a membership certificate.
- $\mathcal{GM}$ records the transcript $\text{transcript}_{\text{id}}$ of Join in reg. We remark that $\text{transcript}_{\text{id}}$ is uniquely determined by id .

$\text{Sign}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), M) \rightarrow \Sigma$. This is a probabilistic algorithm that takes as input group public key gpk , user's secret usk_{id} , user's certificate cert_{id} and a message M . The algorithm returns a signature Σ .

$\text{Verify}(\text{gpk}, M, \Sigma) \rightarrow 1/0$. This deterministic algorithm, on input group public key gpk , a signature Σ and a message M , outputs either 1/0 indicating Σ is valid/invalid.

$\text{Open}(\text{gpk}, \text{osk}, M, \Sigma) \rightarrow \text{id}/\perp$. This algorithm takes public parameters pp , the secret key osk of $\mathcal{OA}$, a *valid* message-signature pair (M, Σ) , outputs a user identifier id or a failure symbol $\perp$.

$\text{Reveal}(\text{gpk}, \text{osk}, \text{id}) \rightarrow \text{trace}_{\text{id}}$. This algorithm is invoked by $\mathcal{OA}$, and returns a tracing trapdoor trace_{id} corresponding to user id .

$\text{Trace}(\text{gpk}, \text{trace}_{\text{id}}, \Sigma) \rightarrow 1/0$. This deterministic algorithm takes as input gpk , a trapdoor trace_{id} , a signature Σ and outputs 1/0.

$\text{Claim}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), (M, \Sigma)) \rightarrow \chi$. This algorithm takes as input gpk , a valid message-signature pair (M, Σ) generated from $(\text{usk}_{\text{id}}, \text{cert}_{\text{id}})$, and outputs a claim χ .

$\text{ClaimVerify}(\text{gpk}, (M, \Sigma), \chi) \rightarrow 1/0$. Given group public key gpk , a valid message-signature pair (M, Σ) and a claim χ , this deterministic algorithm outputs 1 (accept) or 0 (reject).

Definition 7 (Correctness). *The scheme $\mathcal{TS}$ is **correct** if for any message M , for any user id and any signature $\Sigma \leftarrow \text{Sign}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), M)$, the following four conditions are satisfied except with a negligible probability in security parameter λ :*

1. **Sign-correctness:** $\text{Verify}(\text{gpk}, M, \Sigma) = 1$;
2. **Open-correctness:** $\text{Open}(\text{gpk}, \text{osk}, M, \Sigma) = \text{id}$;
3. **Trace-correctness:** $\text{Trace}(\text{gpk}, \text{trace}_{\text{id}}, \Sigma) = 1$ where $\text{trace}_{\text{id}} \leftarrow \text{Reveal}(\text{gpk}, \text{osk}, \text{id})$; and for any Σ' not output from Sign on input $(\text{usk}_{\text{id}}, \text{cert}_{\text{id}})$, we have that $\text{Trace}(\text{gpk}, \text{trace}_{\text{id}}, \Sigma') = 0$;
4. **Claim-correctness:** let $\chi \leftarrow \text{Claim}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), (M, \Sigma))$; then $\text{ClaimVerify}(\text{gpk}, (M, \Sigma), \chi) = 1$.

For security, a TS scheme should be *traceable*, *non-frameable* and *anonymous*. [Appendix A](#) provides the full description of these security notions.

3. A Traceable Signature Scheme from Lattices

3.1. Technical Overview

At a high level, our lattice-based TS scheme is similar to the generic construction of dynamic group signatures in [38]. The building blocks include a digital signature scheme, a CCA-secure public key encryption scheme, and a simulation-sound NIZK proof system. The group manager uses the digital signature scheme to certify membership. The NIZK proof system is invoked by users when they have to prove certain relations in zero knowledge while generating signatures. For these two components, we employ the lattice-based signature scheme by Jeudy *et al.* [44] and the ZK argument system by Yang *et al.* [46]. For the encryption layer, we apply the CHK transformation [51] to GPV-IBE scheme [45] to obtain a CCA-secure encryption.

As explained in Section 1, our construction adapts the revocation technique in [28] for tracing/claiming mechanisms. Recall that the user's tracing trapdoor is a vector $\mathbf{x} \in \mathbb{Z}_q^n$, and the tracing tag will be an LWE sample with $\mathbf{x}$ as secret. In constructions following the KTY model [30], the tracing trapdoor is sent directly to the group manager. However, in the BP model [35], an adversary can passively observe the Join protocol and thus may use the obtained tracing trapdoor to distinguish between users' signatures. To hide the trapdoors, the idea of [35] is to use public key encryption, and to let the group manager certify the encryption of tracing trapdoors instead. However, rather than public key encryption, we can employ a trapdoor one-way function with sufficient pseudorandomness. An LWE-based function suffices: observe that if $\mathbf{x} \in \mathbb{Z}_q^n$ is a tracing key, then a user can "encrypt" $\mathbf{x}$ as $\mathbf{B} \cdot \mathbf{x} + \mathbf{e} \bmod q$ where $\mathbf{e}$ is an error term. Note that by the hardness of LWE, the adversary observing Join learns nothing about $\mathbf{x}$ from $\mathbf{B} \cdot \mathbf{x} + \mathbf{e} \bmod q$. Additionally, $\mathbf{x}$ can be recovered if we know many short vectors $\mathbf{s} \in \mathbb{Z}^{m_B}$ such that $\mathbf{B}^\top \cdot \mathbf{s} = 0 \bmod q$. From this observation, we let $\mathbf{B}$ serve as the public key to the LWE-based one-way function and also as the master public key in GPV-IBE scheme as well. This design choice reduces the size of the opening authority's secret.

When signing a message M , the signer encrypts its unique identifier issued by the group manager and computes the tracing tag as $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e}_t \bmod q$. The matrix $\mathbf{M}$ is freshly sampled per signature by hashing a short randomness ρ . The user then generates an NIZKAoK proving that: 1) the tracing trapdoor is bound to the user's secret, 2) an "encryption" of the tracing trapdoor is certified, 3) the "encryption" of the tracing trapdoor is correct, 4) the encryption of its identifier is correct and 5) the tag is correct. For condition 1), we let the tracing trapdoor $\mathbf{x}$ be the output of an SIS-based one-way function evaluated on the user's secret. We remark that proving condition 1 is crucial to our construction. Otherwise, malicious signers may defeat traceability and non-frameability of the system.

For the NIZK layer, the ZK argument by Yang *et al.* can be adapted to handle the signature-defining relations above, using techniques presented in [61]. To make the argument non-interactive in QROM, we apply Unruh transformation [47, 48].

Supporting ZK argument. The TS scheme uses two prime moduli q and q' . The modulus q , typically

large, is used for the ZK argument of [46] and the lattice-based signature scheme of [44]. In contrast, the modulus q' is much smaller than q and used for the other lattice-based components.

In the construction, we apply ZK argument by Yang *et al.* [46] to prove the knowledge of a tuple $(\text{id}, \mathbf{z}, \mathbf{x}, \mathbf{e}, \text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2), \mathbf{r}_{\mathbf{c}}, \mathbf{e}_{\mathbf{t}})$ such that:

- (i) $(\mathbf{z}, \mathbf{x})$ is a preimage-image pair of an SIS-based one-way function;
- (ii) id and $(\mathbf{v}_1, \mathbf{v}_2)$ form a tag-signature pair on $\text{bin}(\mathbf{y})$ under the signature scheme of [44]; here $\text{bin}(\mathbf{y})$ is the binary decomposition of $\mathbf{y}$ along a gadget matrix;
- (iii) $\mathbf{y}$ is an LWE-sample generated from secret $\mathbf{x}$ with error $\mathbf{e}$;
- (iv) A GPV-IBE ciphertext $\mathbf{c}$ is an encryption of $\text{bin}(\text{id})$, with randomnesses $\mathbf{r}_{\mathbf{c}}$;
- (v) An LWE sample $\mathbf{t}$ is generated from the secret $\mathbf{x}$ and error $\mathbf{e}_{\mathbf{t}}$.

Proving statement (ii) is equivalently to proving knowledge of message-signature pair in the scheme from [44] and thus can be handled by the ZK framework from [46]. We briefly explain how to prove in ZK the statements (i), (iii), (iv) and (v). In our construction, statements (i), (iii), (iv), (v) consist of equation modulo q' , while the ZK system works with a different modulo q . Therefore, we need to appropriately transform these statements into some sets of equations modulo q . Moreover, we observe that statements (i), (iii), (iv), (v) are special cases of the following relation

$$\mathcal{R}_{m,n,q',\beta} = \{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_{q'}^{m \times n} \times \mathbb{Z}_{q'}^m \times \mathbb{Z}_{q'}^n : \mathbf{y} = \mathbf{A} \cdot \mathbf{x} \bmod q'; \|\mathbf{x}\|_{\infty} \leq \beta\},$$

which, by following the binary decomposition technique of [61], can be transformed to the following linear relation with binary constraint over the witness

$$\mathcal{R}_{\text{bin}} = \{((\overline{\mathbf{A}'}, \mathbf{y}'), \overline{\mathbf{x}'}) \in \mathbb{Z}_q^{m' \times n'} \times \mathbb{Z}_q^{n'} \times \{0, 1\}^{m'} : \mathbf{y}' = \overline{\mathbf{A}'} \cdot \overline{\mathbf{x}'} \bmod q\}.$$

The above relation can be proved in ZK using the ZK argument of [46]. We devote Appendix D.2 for the detailed transformations applied for each of the statements (i), (iii), (iv), (v) to a case of $\mathcal{R}_{\text{bin}}$.

3.2. Description of the Scheme

Let λ be a security parameter and let $n = \Theta(\lambda)$ be a lattice dimension. Let $N = 2^\ell - 1 \in \text{poly}(\lambda)$ be the maximum group size, which also defines the identity space $[N]$. The public parameter pp output from Setup consists of parameters and distributions defined from λ, n and N . They are set so that all algorithms run in polynomial time and are correct with overwhelming probability. Furthermore, the parameter regimes defining the LWE and the SIS problems, on which security of the scheme is based on, are secure against known attacks. The parameters and distributions are as follows:

- Prime moduli $q, q' \in \text{poly}(\lambda)$ such that $q > N$;
- Dimension $m_{\mathbf{F}} = n \lceil \log q' \rceil + \Theta(\lambda)$;
- Dimension $m_{\mathbf{B}} = 2n \lceil \log q' \rceil + \Theta(\lambda)$;
- Dimensions m_1, m_2 such that $m_1 \log 3 = n \log q + \Theta(\lambda)$ and $m_2 = n \lceil \log q \rceil$;
- Dimension $m_{\mathbf{M}} = 3n$;
- Gaussian widths $\sigma_{\text{sign}}, \sigma_{\text{com}} > 0$; these parameters are the preimage sampling widths and commitment randomness sampling widths in the signing algorithm of the signature scheme of [44]. In addition, let $\sigma_{\text{verif}} = \sqrt{\sigma_{\text{com}}^2 + \sigma_{\text{sign}}^2}$;
- ℓ_{∞} -norm bounds $\beta_1 = \sigma_{\text{verif}} \log m_1$ and $\beta_2 = \sigma_{\text{sign}} \log m_2$;
- Parameters α_{GPV} and $\sigma_{\text{GPV}} > \sqrt{n + \log m_{\mathbf{B}}}$. The former defines a discrete Gaussian distribution $\mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}$ for sampling GPV-IBE encryption randomness, the latter defines a discrete Gaussian distribution $\mathcal{D}_{\sigma_{\text{GPV}}}$ for sampling decryption key. In our scheme, GPV-IBE encrypts messages in $\{0, 1, \dots, N\}$, therefore we require that $\alpha_{\text{GPV}} \cdot q' \cdot \sqrt{n} \cdot \sigma_{\text{GPV}} \cdot \sqrt{m_{\mathbf{B}}} + 1 < q' / (4(N + 1))$ for decryption to be correct. We also let $B_{\text{GPV}} = \alpha_{\text{GPV}} \cdot q' \cdot \sqrt{m_{\mathbf{B}}} + 1$;
- A distribution D_{LWE} over $\mathbb{Z}$ such that elements sampled from D_{LWE} has absolute values upper bounded by B_{LWE} with overwhelming probability in λ . We require $(4B_{\text{LWE}} + 1)^2 < q'$ and $\sigma_{\text{GPV}} \cdot \sqrt{m_{\mathbf{B}}} \cdot B_{\text{LWE}} < q/2$;
- Parameters defining the public key to the BDLOP commitment scheme [62], which serves as the common reference string crs to the ZK argument of [46];
- A repetition parameter $\kappa = \Theta(\lambda / \log \lambda)$, here $p \in \text{poly}(\lambda)$ is a small integer determining the soundness error $2/(2p + 1)$ of the ZK argument of [46];

The public parameter pp is given as

$$\text{pp} = (n, q, q', m_{\mathbf{F}}, m_{\mathbf{B}}, m_1, m_2, m_{\mathbf{M}}, \sigma_{\text{sign}}, \sigma_{\text{com}}, \alpha_{\text{GPV}}, \sigma_{\text{GPV}}, D_{\text{LWE}}, \text{crs}, \kappa).$$

To sign a message, signer needs to generate an NIZKAoK for the following relation:

Definition 8. *Define*

$$\mathcal{R}_{\text{Sign}} = \left\{ \left((\mathbf{A}, \mathbf{A}', \mathbf{D}, \mathbf{u}, \mathbf{F}, \mathbf{B}, \mathbf{v}, \mathbf{M}, \mathbf{c}, \mathbf{t}), (\text{id}, \mathbf{z}, \mathbf{x}, \mathbf{e}, \text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2), \mathbf{r}, \mathbf{e}_{\mathbf{c}}, \mathbf{e}_{\mathbf{t}}) \right) \right\}$$

as a relation, where

$$(i) \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m_1}, \mathbf{A}' \in \mathbb{Z}_q^{n \times m_2}, \mathbf{D} \in \mathbb{Z}_q^{n \times m_{\mathbf{B}} \lceil \log q' \rceil}, \mathbf{B} \in \mathbb{Z}_{q'}^{n \times m_{\mathbf{B}}}, \mathbf{v} \in \mathbb{Z}_{q'}^n, \mathbf{F} \in \mathbb{Z}_{q'}^{n \times m_{\mathbf{F}}}, \mathbf{M} \in \mathbb{Z}_{q'}^{m_{\mathbf{M}} \times n}, \mathbf{u} \in \mathbb{Z}_q^n, \\ \mathbf{c} \in \mathbb{Z}_{q'}^{m_{\mathbf{B}} + \ell}, \mathbf{t} \in \mathbb{Z}_{q'}^{m_{\mathbf{M}}};$$

- (ii) $\text{id} \in \{1, \dots, N\}$, $\mathbf{z} \in \{0, 1\}^{m_F}$, $\mathbf{x} \in \mathbb{Z}_{q'}^n$, $\mathbf{e} \in [-B_{\text{LWE}}, B_{\text{LWE}}]^{m_B}$, $\mathbf{y} \in \mathbb{Z}_{q'}^{m_B}$, $\mathbf{v}_1 \in [-\beta_1, \beta_1]^{m_1}$, $\mathbf{v}_2 \in [-\beta_2, \beta_2]^{m_2}$, $\mathbf{r} \in \mathbb{Z}_{q'}^n$, $\mathbf{e}_c \in [-B_{\text{GPV}}, B_{\text{GPV}}]^{m_B+1}$; $\mathbf{e}_t \in [-B_{\text{LWE}}, B_{\text{LWE}}]^{m_M}$;
- (iii) $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$;
- (iv) $[\mathbf{A} \mid \text{id} \cdot \mathbf{G}_n + \mathbf{A}'] \cdot (\mathbf{v}_1, \mathbf{v}_2)^\top = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{y}) \bmod q$, where $\mathbf{G}_n \in \mathbb{Z}_q^{n \times m_2}$ is a gadget matrix;
- (v) $\mathbf{B}^\top \cdot \mathbf{x} + \mathbf{e} = \mathbf{y} \bmod q'$;
- (vi) $\begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r} + \mathbf{e}_c + \begin{pmatrix} \mathbf{0}^n \\ \lceil q'/2(N+1) \rceil \cdot \text{id} \end{pmatrix} = \mathbf{c} \bmod q'$;
- (vii) $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e}_t \bmod q'$.

By the techniques discussed in [Appendix D.2](#), $\mathcal{R}_{\text{Sign}}$ can be transformed into a case of the following relation

$$\{((\mathbf{A}, \mathbf{y}, \mathcal{S}), \mathbf{x}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q \wedge \forall (h, i, j) \in \mathcal{S} : \mathbf{x}[h] = \mathbf{x}[i] \cdot \mathbf{x}[j] \bmod q\},$$

where $\mathbf{A} \in \mathbb{Z}_q^{m'+n'}$, $\mathbf{y} \in \mathbb{Z}_q^{m'}$ and $\mathcal{S}$ is a set of 3-tuples of size s each of which consists of 3 integers in $[1, n']$. By applying the Unruh transformation to the ZK argument [46], we obtain an NIZKAoK for $\mathcal{R}_{\text{Sign}}$.

In addition, as the signer claims a signature by generating an NIZKAoK, the relation signer needs to prove is defined as follows:

Definition 9. *Define*

$$\mathcal{R}_{\text{Claim}} = \left\{ \left((\mathbf{F}, \mathbf{M}, \mathbf{t}), (\mathbf{z}, \mathbf{x}, \mathbf{e}_t) \right) \right\}$$

as a relation, where

- (i) $\mathbf{F} \in \mathbb{Z}_{q'}^{n \times m_F}$, $\mathbf{M} \in \mathbb{Z}_{q'}^{m_M \times n}$, $\mathbf{t} \in \mathbb{Z}_{q'}^{m_M}$;
- (ii) $\mathbf{z} \in \{0, 1\}^{m_F}$, $\mathbf{x} \in \mathbb{Z}_{q'}^n$, $\mathbf{e} \in [-B_{\text{LWE}}, B_{\text{LWE}}]^{m_B}$;
- (iii) $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$;
- (iv) $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e}_t \bmod q'$.

Similarly, we obtain an NIZKAoK for $\mathcal{R}_{\text{Claim}}$ by applying Unruh transformation to the ZK argument [46].

Now we describe the remaining algorithms in our lattice-based TS scheme:

- **KeyGen**(1^λ) $\rightarrow$ (gpk, gsk, osk, reg): this algorithm performs the following steps.

1. Generate verification key

$$(\mathbf{A}, \mathbf{A}', \mathbf{D}, \mathbf{u}) \in \mathbb{Z}_q^{n \times m_1} \times \mathbb{Z}_q^{n \times m_2} \times \mathbb{Z}_q^{n \times m_B \lceil \log q' \rceil} \times \mathbb{Z}_q^n$$

and signing key $\mathbf{R}_A$ for the signature scheme of [44] as follows:

- Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m_1}$;
 - Sample $\mathbf{R}_\mathbf{A} \xleftarrow{\$} \{-1, 0, 1\}^{m_1 \times m_2}$;
 - Compute $\mathbf{A}' = -\mathbf{A} \cdot \mathbf{R}_\mathbf{A} \bmod q$;
 - Sample $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_q^{n \times m_\mathbf{B} \lceil \log q' \rceil}$.
2. Sample matrices $\overline{\mathbf{B}} \xleftarrow{\$} \mathbb{Z}_{q'}^{n \times (m_\mathbf{B} - n \lceil \log q' \rceil)}$ and $\mathbf{S}_\mathbf{B} \xleftarrow{\$} \{0, 1\}^{(m_\mathbf{B} - n \lceil \log q' \rceil) \times n \lceil \log q' \rceil}$. Restart if $\overline{\mathbf{B}}$ is not full-rank. Define $\mathbf{B} = (\overline{\mathbf{B}} | \mathbf{G}_n - \overline{\mathbf{B}} \cdot \mathbf{S}_\mathbf{B}) \in \mathbb{Z}_{q'}^{n \times m_\mathbf{B}}$;
 3. Sample $\mathbf{F} \xleftarrow{\$} \mathbb{Z}_{q'}^{n \times m_\mathbf{F}}$ that defines a SIS-based one-way function.
 4. Select a strongly secure one-time signature scheme $\mathcal{OTS} = (\text{KeyGen}, \text{Sign}, \text{Verif})$. Let $l = \text{poly}(\lambda)$ be the bit length of the verification key output by $\mathcal{OTS}.\text{KeyGen}$.
 5. Select hash functions $\mathcal{H}_{\text{GPV}} : \{0, 1\}^l \rightarrow \mathbb{Z}_{q'}^n$, $\mathcal{H}_{\text{LWE}} : \{0, 1\}^\lambda \rightarrow \mathbb{Z}_{q'}^{m_\mathbf{M} \times n}$, $\mathcal{H}_{\text{Sign}}^{(1)} : \{0, 1\}^* \rightarrow \{1, 2, 3, 4\}^\kappa$, $\mathcal{H}_{\text{Sign}}^{(2)} : D_{\text{Sign}} \rightarrow D_{\text{Sign}}$, $\mathcal{H}_{\text{Claim}}^{(1)} : \{0, 1\}^* \rightarrow \{1, 2, 3, 4\}^\kappa$, $\mathcal{H}_{\text{Claim}}^{(2)} : D_{\text{Claim}} \rightarrow D_{\text{Claim}}$ that serve as our random oracles. Here D_{Sign} and D_{Claim} denotes the set of all prover's responses in the underlying ZK system for $\mathcal{R}_{\text{Sign}}$ and $\mathcal{R}_{\text{Claim}}$ respectively. Looking forward, $\mathcal{H}_{\text{Sign}}^{(1)}, \mathcal{H}_{\text{Sign}}^{(2)}, \mathcal{H}_{\text{Claim}}^{(1)}$ and $\mathcal{H}_{\text{Claim}}^{(2)}$ are used in Unruh transformation to convert the ZK arguments into non-interactive versions.

Output $\text{gsk} = \mathbf{R}_\mathbf{A}$, $\text{osk} = \mathbf{S}_\mathbf{B}$, $\text{reg} = \epsilon$ and

$$\text{gpk} = (\mathbf{A}, \mathbf{A}', \mathbf{D}, \mathbf{u}, \mathbf{B}, \mathbf{F}, \mathcal{OTS}, \mathcal{H}_{\text{GPV}}, \mathcal{H}_{\text{LWE}}, \mathcal{H}_{\text{Sign}}^{(1)}, \mathcal{H}_{\text{Sign}}^{(2)}, \mathcal{H}_{\text{Claim}}^{(1)}, \mathcal{H}_{\text{Claim}}^{(2)});$$

- Join $\langle \mathcal{GM}(\text{gsk}), \mathcal{U} \rangle$ (gpk): group manager $\mathcal{GM}$ initializes a counter st at 0. A prospective user $\mathcal{U}$, who wants to join the group, interacts with $\mathcal{GM}$ following these steps.

1. The user selects $\mathbf{z} \xleftarrow{\$} \{0, 1\}^{m_\mathbf{F}}$, computes $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q' \in \mathbb{Z}_{q'}^n$, samples $\mathbf{e} \leftarrow D_{\text{LWE}}^m$ and computes $\mathbf{y} = \mathbf{B}^\top \cdot \mathbf{x} + \mathbf{e} \bmod q' \in \mathbb{Z}_{q'}^{m_\mathbf{B}}$, then signs $\text{bin}(\mathbf{y}) \in \{0, 1\}^{m_\mathbf{B} \lceil \log q' \rceil}$ using an ordinary signature. The message $\text{bin}(\mathbf{y})$ and the signature σ are sent to $\mathcal{GM}$;
2. $\mathcal{GM}$ checks the registry table whether $\text{bin}(\mathbf{y})$ is not previously certified registered and σ is valid. If both checks are successful, $\mathcal{GM}$ increases st by 1 and assigns the identifier $\text{id} = \text{st}$. It then signs $\text{bin}(\mathbf{y}) \in \{0, 1\}^{m_\mathbf{B} \lceil \log q' \rceil}$ under the signature scheme of [44], using $\text{gsk} = \mathbf{R}_\mathbf{A}$ as signing key and id as tag. The signature has the form $(\text{id}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_q \setminus \{0\} \times [-\beta_1, \beta_1]^{m_1} \times [-\beta_2, \beta_2]^{m_2}$, and satisfies

$$[\mathbf{A} \mid \text{id} \cdot \mathbf{G}_n + \mathbf{A}'] \cdot (\mathbf{v}_1, \mathbf{v}_2) = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{y}) \bmod q. \quad (1)$$

$\mathcal{GM}$ then sends $(\text{id}, \mathbf{v}_1, \mathbf{v}_2)$ to user $\mathcal{U}$;

3. User $\mathcal{U}$ verifies $(\text{id}, \mathbf{v}_1, \mathbf{v}_2)$ by checking (1). It sets $\text{usk}_{\text{id}} = \mathbf{z} \in \{0, 1\}^{m_{\mathbf{F}}}$ and

$$\text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}} \times [-\beta_1, \beta_1]^{m_1} \times [-\beta_2, \beta_2]^{m_2};$$

4. $\mathcal{GM}$ sets $\text{transcript}_{\text{id}} = (\text{id}, \sigma, \text{cert}_{\text{id}})$ and updates the registry table $\text{reg} := \text{reg} \parallel \text{transcript}_{\text{id}}$.

- $\text{Sign}(\text{gpk}, \text{usk}_{\text{id}}, \text{cert}_{\text{id}}, M)$: user id executes this algorithm on input gpk , secret $\text{usk}_{\text{id}} = \mathbf{z} \in \{0, 1\}^{m_{\mathbf{F}}}$, certificate $\text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}} \times [-\beta_1, \beta_1]^{m_1} \times [-\beta_2, \beta_2]^{m_2}$ and a message $M \in \{0, 1\}^*$. It performs these steps:

1. Compute $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$;
2. Compute $\mathbf{e} = \mathbf{y} - \mathbf{B}^\top \cdot \mathbf{x} \bmod q'$. Note that $\|\mathbf{e}\|_\infty \leq B_{\text{LWE}}$;
3. Generate a key pair $(vk, sk) \leftarrow \mathcal{OTS}.\text{KeyGen}(1^\lambda)$. Then encrypt id with respect to "identity" vk by computing $\mathbf{v} = \mathcal{H}_{\text{GPV}}(vk) \in \mathbb{Z}_q^n$; sampling $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_{\mathbf{c}} \leftarrow \mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}^{m_{\mathbf{B}}+1}$. Let the final ciphertext be

$$\mathbf{c} = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r} + \mathbf{e}_{\mathbf{c}} + \begin{pmatrix} \mathbf{0}^{m_{\mathbf{B}}} \\ \lceil q'/2(N+1) \rceil \cdot \text{id} \end{pmatrix} \bmod q';$$

4. Sample randomness $\rho \xleftarrow{\$} \{0, 1\}^\lambda$ and compute $\mathbf{M} = \mathcal{H}_{\text{LWE}}(\rho) \in \mathbb{Z}_{q'}^{m_{\mathbf{M}} \times n}$, sample $\mathbf{e}_{\mathbf{t}} \leftarrow D_{\text{LWE}}^m$ and set $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e}_{\mathbf{t}} \bmod q'$;
5. Using the witness $(\text{id}, \mathbf{z}, \mathbf{x}, \mathbf{e}, \text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2), \mathbf{r}, \mathbf{e}_{\mathbf{c}}, \mathbf{e}_{\mathbf{t}})$, generate a "signature of knowledge" (an NIZKAoK) π for the relation $\mathcal{R}_{\text{Sign}}$ (Definition 8), on the message M . This is done by applying the Unruh transformation for generalized Σ -protocol [48] using hash functions $\mathcal{H}_{\text{Sign}}^{(1)}$ and $\mathcal{H}_{\text{Sign}}^{(2)}$. Specifically, when applying the transformation, it computes

$$\mathcal{H}_{\text{Sign}}^{(1)}(\text{crs}, M, \rho, x, (\text{com}_i)_{[\kappa]}, (\text{ch}_{i,j})_{[\kappa] \times [4]}, (h_{i,j})_{[\kappa] \times [4]}),$$

where x is the statement of $\mathcal{R}_{\text{Sign}}$, com_i ; and $\text{ch}_{i,j}, h_{i,j} = \mathcal{H}_{\text{Sign}}^{(2)}(\text{rsp}_{i,j})$ denote the commitment, challenge, hash of response generated by the prover. Let π be the final proof.

6. Compute a one-time signature $\text{sig} \leftarrow \mathcal{OTS}.\text{Sign}(sk, (\rho, \mathbf{c}, \mathbf{t}, \pi))$;

The algorithm outputs the signature

$$\Sigma = (\rho, \mathbf{c}, \mathbf{t}, \pi, vk, \text{sig}). \quad (2)$$

- $\text{Verify}(\text{gpk}, M, \Sigma)$: this algorithm parses the input signature Σ as in (2). The algorithm returns 1 if and only if $\mathcal{OTS}.\text{Verif}(vk, (\rho, \mathbf{c}, \mathbf{t}, \pi), \text{sig}) = 1$ and π is a valid NIZKAoK for $\mathcal{R}_{\text{Sign}}$ as in Definition 8.

- **Open**(gpk, osk = $\mathbf{S}_B$, M , Σ): return $\perp$ if Σ does not verify with M . Otherwise, the algorithm proceeds as follows:
 1. Compute $\mathbf{v} = \mathcal{H}_{\text{GPV}}(vk) \in \mathbb{Z}_{q'}^n$. Next, using $\mathbf{S}_B$, sample $\mathbf{e}_{vk} \leftarrow \mathcal{D}_{\sigma_{\text{GPV}}}^{m_B}$ such that $\mathbf{B} \cdot \mathbf{e}_{vk} = \mathbf{v} \bmod q'$. This is done by the algorithm of Proposition 1;
 2. Decrypt $\mathbf{c}$ by computing $\lceil (-\mathbf{e}_{vk}^\top | 1) \cdot \mathbf{c}_1 \bmod q' \rceil / (q'/2(N+1))$. Let $\text{id} \in \{0, \dots, N\}$ be the result, output id .
- **Reveal**(gpk, osk, id): on input an identifier $\text{id} \in \{1, \dots, N\}$ and $\text{osk} = \mathbf{S}_B$, the algorithm checks the public registry table for an entry starting with id . If such an entry does not exist, output $\perp$. Otherwise, it retrieves the component $\mathbf{y} \in \mathbb{Z}_{q'}^{m_B}$ in cert_{id} and inverts the LWE sample $\mathbf{y}$ as follows:
 1. Using the algorithm of Proposition 1, sample a set of $\mathbb{R}$ -linearly independent vectors $\mathbf{s}_1, \dots, \mathbf{s}_{m_B}$ from $\mathcal{D}_{\sigma_{\text{GPV}}}^{m_B}$ such that $\mathbf{B} \cdot \mathbf{s}_i = \mathbf{0} \bmod q$ for $i = 1, \dots, m_B$. Let $\mathbf{S} = (\mathbf{s}_1 | \dots | \mathbf{s}_{m_B}) \in \mathbb{Z}^{m_B \times m_B}$. Note that by Theorem 6, the algorithm of Proposition 1 should be invoked $\mathcal{O}(m_B^2)$ times so that the $\mathbf{s}_i$'s can be found;
 2. Compute $\mathbf{S}^\top \cdot \mathbf{y} \bmod q'$. Note that if $\mathbf{y} = \mathbf{B}^\top \cdot \mathbf{x} + \mathbf{e} \bmod q'$, then $\mathbf{S}^\top \cdot \mathbf{y} \bmod q' = \mathbf{S}^\top \cdot \mathbf{e} \bmod q' = \mathbf{S}^\top \cdot \mathbf{e}$. This is because $\|\mathbf{S}^\top \cdot \mathbf{e}\|_2 \leq \|\mathbf{S}\|_2 \cdot \|\mathbf{e}\|_\infty \leq \sigma_{\text{GPV}} \cdot \sqrt{m_B} \cdot B_{\text{LWE}} < q/2$;
 3. Using linear algebra, solve the linear equation $\mathbf{S}^\top \cdot \mathbf{e} = \mathbf{S}^\top \cdot \mathbf{y} \bmod q'$ over $\mathbb{Z}$. Output $\perp$ if $\mathbf{e}$ does not exist or $\|\mathbf{e}\|_\infty > B_{\text{LWE}}$;
 4. Using linear algebra, solve the linear equation $\mathbf{B}^\top \cdot \mathbf{x} = \mathbf{y} - \mathbf{e} \bmod q'$ over $\mathbb{Z}_{q'}$. Note that as $\mathbf{B}^\top$ is full-rank, there exists at most one solution $\mathbf{x}$;
 5. Output $\text{trace}_{\text{id}} = \mathbf{x}$. If no such $\mathbf{x}$ exists then output $\perp$.
- **Trace**(gpk, trace_{id} , Σ): on input gpk, trapdoor $\text{trace}_{\text{id}} = \mathbf{x} \in \mathbb{Z}_{q'}^n$ and the signature Σ , the algorithm parses Σ as in (2) and computes $\mathbf{M} = \mathcal{H}_{\text{LWE}}(\rho)$. Returns 1 iff $\|(\mathbf{t} - \mathbf{M} \cdot \mathbf{x} \bmod q')\|_\infty \leq B_{\text{LWE}}$.
- **Claim**(gpk, usk_{id} , cert_{id} , (M, Σ)): The algorithm takes as input public key gpk, secret $\text{usk}_{\text{id}} = \mathbf{z}$, certificate $\text{cert}_{\text{id}} = (\mathbf{y}, \mathbf{v}_1, \mathbf{v}_2)$ and a message-signature pair (M, Σ) . It parses Σ as in (2), computes $\mathbf{M} = \mathcal{H}_{\text{LWE}}(\rho)$, $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$ and $\mathbf{e}_t = \mathbf{t} - \mathbf{M} \cdot \mathbf{x} \bmod q'$. Returns $\perp$ if $\|\mathbf{e}_t\|_\infty > B_{\text{LWE}}$. Else, it generates an NIZKAoK χ for the relation $\mathcal{R}_{\text{Claim}}$ defined in Definition 9, using witness $(\mathbf{x}, \mathbf{z}, \mathbf{e}_t)$. Similar to Sign, the NIZKAoK is obtained via the Unruh transformation with random oracles $\mathcal{H}_{\text{Claim}}^{(1)}$ and $\mathcal{H}_{\text{Claim}}^{(2)}$. When applying the transformation, it computes

$$\mathcal{H}_{\text{Claim}}^{(1)}(\text{crs}, M, \rho, x, (\text{com}_i)_{[\kappa]}, (\text{ch}_{i,j})_{[\kappa] \times [4]}, (h_{i,j})_{[\kappa] \times [4]}),$$

where x is the statement of $\mathcal{R}_{\text{Claim}}$; and $\text{com}_i, \text{ch}_{i,j}, h_{i,j} = \mathcal{H}_{\text{Claim}}^{(2)}(\text{rsp}_{i,j})$ are commitment, challenge, hash of response generated by the prover. Let χ be the final proof. The algorithm outputs χ .

- $\text{ClaimVerify}(\text{gpk}, (M, \Sigma), \chi)$: the algorithm takes as input gpk , a valid message-signature pair (M, Σ) and a claim χ . It parses Σ as in (2), then from gpk and Σ it reconstructs the statement $x \in \mathcal{R}_{\text{Claim}}$. It outputs 1 if the claim χ is a valid NIZKAoK for x , and 0 otherwise.

4. Analysis of the Scheme

4.1. Correctness and Efficiency

4.1.1. Correctness

The sign/open/claim-correctness of the TS scheme in Section 3.2 follows from the correctness of its building blocks: the lattice-based signature scheme of [44], GPV-IBE scheme [45], the ZK argument system of [46] and the NIZK from generalized Unruh transformation [48]. Tracing correctness is implied by the following lemma.

Lemma 10. *Let $\text{trace}_{\text{id}} = \mathbf{x}, \text{trace}_{\text{id}'} = \mathbf{x}' \in \mathbb{Z}_q^n$ be the output of Reveal with respect to different users $\text{id}, \text{id}' \in \{1, \dots, N\}$. Assume that Σ is a signature created by id , then $\text{Trace}(\text{gpk}, \text{trace}_{\text{id}'}, \Sigma) = 1$ with negligible probability.*

PROOF. First note that Reveal correctly recovers the tracing trapdoors $\mathbf{x}$ and $\mathbf{x}'$ with overwhelming probability. In addition, $\mathbf{x}$ and $\mathbf{x}'$ are independent and distributed statistically close to uniform (by Lemma 5). Therefore we have that $\mathbf{x} \neq \mathbf{x}'$ except with probability at most $2^{-\lambda} + q'^{-n} = \text{negl}(\lambda)$.

Parse the signature Σ as in (2). The condition $\text{Trace}(\text{gpk}, \text{trace}_{\text{id}'}, \Sigma) = 1$ is equivalent to $\|\mathbf{t} - \mathbf{M} \cdot \mathbf{x}'\| \leq B_{\text{LWE}}$. It follows that

$$\|\mathbf{M} \cdot (\mathbf{x} - \mathbf{x}')\|_{\infty} \leq \|\mathbf{t} - \mathbf{M} \cdot \mathbf{x}\|_{\infty} + \|\mathbf{t} - \mathbf{M} \cdot \mathbf{x}'\|_{\infty} \leq 2B_{\text{LWE}}.$$

Since $\mathbf{M}$ is uniformly distributed, the above only happens with probability at most $q'^{-n} = \text{negl}(\lambda)$, by Lemma 4. $\square$

4.1.2. Efficiency

We analyze the asymptotic efficiency of the proposed scheme regarding key and signature sizes. The size of gpk is dominated by the size of verification key of the signature scheme from [44]. The user secret key usk_{id} is a binary vector of length $m_{\mathbf{F}} = \mathcal{O}(n \log q') = \mathcal{O}(\lambda \log \lambda)$. The group manager and opening authority secrets are matrices with bit sizes of order $\mathcal{O}((n \log q)^2) = \mathcal{O}(\lambda^2 \log^2 \lambda)$ and $\mathcal{O}((n \log q')^2) = \mathcal{O}(\lambda^2 \log^2 \lambda)$ respectively.

To estimate the signature size, we observe that in a signature Σ of the form in (2), the most dominant parts are the verification key vk and the signature sig in the one-time signature scheme $\mathcal{OTS}$. In turn, the size of vk and the size of sig depend on the bit-length $l_{\mathcal{OTS}}$ of the message that is signed under $\mathcal{OTS}$.

Note that we employ $\mathcal{OTS}$ to sign the randomness ρ , the ciphertext $\mathbf{c}$ encrypting id , the tag $\mathbf{t}$ and the NIZK proof π altogether. Among these four components, the bit-size of π dominates and is 4κ times the size of prover's response in the ZK argument [46] for $\mathcal{R}_{\text{Sign}}$. As the size of prover's response is $\mathcal{O}(\lambda \log^3 \lambda)$ (see Appendix D.3) and $\kappa = \Theta(\lambda / \log \lambda)$, it follows that the size of π is $\mathcal{O}(\lambda^2 \log^2 \lambda)$. By instantiating $\mathcal{OTS}$ with the lattice-based one-time signature in [63], we obtain one-time signature size of order $\mathcal{O}(\lambda \log \lambda)$ and one-time verification key size of order $\mathcal{O}(\lambda^2 \log^2 \lambda + \lambda \log \lambda \cdot l_{\mathcal{OTS}})$. Therefore, we obtain a traceable signature scheme with signature size of order $\mathcal{O}(\lambda^3 \log^3 \lambda)$.

For the size of a claim, we note that the size of prover's response in the ZK argument [46] for $\mathcal{R}_{\text{Claim}}$ is of order $\mathcal{O}(\lambda \log^2 \lambda)$ (see Appendix D.3). Since the size of the claim is 4κ times the size of prover's response in the ZK argument [46] for $\mathcal{R}_{\text{Claim}}$, it follows that the bit-size of a claim in our traceable scheme is of order $\mathcal{O}(\lambda^2 \log \lambda)$.

4.2. Security

The security of our TS scheme is stated in the following theorem

Theorem 11. *Suppose that SIS and LWE assumptions hold. Then the lattice-based TS scheme presented in Section 3.2 is CCA-anonymous, traceable and non-frameable in QROM.*

The proof of Theorem 11 relies on a result in [64], which states that polynomial and truly random functions are perfectly indistinguishable against PPT quantum adversaries.

Proposition 2 ([64, 48]). *A uniformly random polynomial function of the degree at least $2q - 1$ is perfectly indistinguishable from a random function for any PPT quantum algorithm performing at most q queries.*

Proposition 2 is particularly useful for proofs in QROM. It provides a way to perfectly simulate a quantum random oracle, which is crucial for the zero-knowledge and simulation-sound online-extractable of NIZK from generalized Unruh transformation [48].

We prove Theorem 11 by proving a sequence of lemmas. The first is as follows.

Lemma 12 (Anonymity). *Let $N \in \text{poly}(\lambda)$ be the maximum group size. Let $q_{\text{Sign}}, q_{\text{Open}}, q_{\mathcal{H}_{\text{GPV}}} \in \text{poly}(\lambda)$ be the maximum numbers of adversary's queries to $\mathcal{O}_{\text{Sign}}$, $\mathcal{O}_{\text{Open}}$ and $\mathcal{H}_{\text{GPV}}$ respectively. Suppose that*

- (i) *The $\text{LWE}_{n, q', D_{\text{LWE}}}$ assumption holds;*
- (ii) *The $\text{LWE}_{n, q', \mathcal{D}_{\alpha, q'}}$ assumption holds, for α satisfying $\alpha_{\text{GPV}}/2\alpha > \sqrt{m_{\mathbf{B}} \cdot \sigma_{\text{GPV}}^2 + 1}$;*
- (iii) *$\mathcal{OTS}$ is a strongly secure one-time signature scheme;*
- (iv) *The employed NIZK argument is zero-knowledge.*

Then the TS scheme presented in Section 3.2 is CCA-anonymous against any PPT adversary in the QROM.

PROOF. We consider a sequence of games. The first game is the experiment $\text{Exp}_{\mathcal{T}\mathcal{S},\mathcal{A}}^{\text{anon}}$ defining CCA-anonymity (see Figure A.3) executed between a PPT adversary $\mathcal{A}$ and a challenger. The last game is the experiment $\text{Exp}_{\mathcal{T}\mathcal{S},\mathcal{A}}^{\text{anon}}$, modified so that the advantage of $\mathcal{A}$ is negligible. In addition, we let W_i be the probability that $\text{Exp}_{\mathcal{T}\mathcal{S},\mathcal{A}}^{\text{anon}}$ returns 1, and the advantage of $\mathcal{A}$ in **Game- i** is $\text{Adv}_i = |\Pr[W_i] - 1/2|$. We argue that all the games are indistinguishable.

Game-0: This is the original $\text{Exp}_{\mathcal{T}\mathcal{S},\mathcal{A}}^{\text{anon}}$. In this game, the challenger generates $(\text{gpk}, \text{gsk}, \text{osk}, \text{reg} = \epsilon)$ by running **KeyGen** and gives (gpk, gsk) to the adversary $\mathcal{A}$. With the oracles provided in the experiment, $\mathcal{A}$ outputs a tuple $(M^*, \text{id}_0^*, \text{id}_1^*)$. The challenger checks $\text{id}_0^*, \text{id}_1^*$ as in line 3 in Figure A.3, then chooses a uniformly random bit b and outputs the challenge signature Σ^* by running $\text{Sign}(\text{gpk}, \text{usk}_{\text{id}_b^*}, \text{cert}_{\text{id}_b^*})$. Note that, Σ^* has the form

$$\Sigma^* = (\rho^*, \mathbf{c}^*, \mathbf{t}^*, \pi^*, vk^*, sig^*)$$

as in (2). Adversary $\mathcal{A}$ is still allowed oracle accesses except for trivial reveal/claim/opening queries. Finally, $\mathcal{A}$ outputs a bit b' and wins the game if $b' = b$. Obviously, $\text{Adv}_0 = \text{Adv}_{\mathcal{T}\mathcal{S},\mathcal{A}}^{\text{anon}}$. We note that at the start of **Game-0**, the random oracles are chosen as truly random functions.

Game-1: In this game, the challenger chooses uniformly at random an unordered pair of identifiers $(\text{id}'_0, \text{id}'_1)$ at the beginning of the experiment. This is a guess for the targeted identifiers $(\text{id}_0^*, \text{id}_1^*)$ that will be chosen by $\mathcal{A}$. The challenger then interacts with $\mathcal{A}$ faithfully. When $\mathcal{A}$ submits its challenge identifiers, challenger checks if the guess is correct. If not, the challenger aborts. Else, it continues the experiment.

Obviously, **Game-1** is identical to **Game-0** with the addition that the challenger guesses the targeted identifiers. The guess is correct with a probability at least $2/(N(N-1))$, since this is independent with $\mathcal{A}$'s view. It follows that $\text{Adv}_1 \geq 2/(N(N-1)) \cdot \text{Adv}_0$, or $\text{Adv}_0 \leq N(N-1)/2 \cdot \text{Adv}_1$.

Game-2: In this game, we change how the challenge signature is generated. Recall that the challenge signature has the form $\Sigma^* = (\rho^*, \mathbf{c}^*, \mathbf{t}^*, \pi^*, vk^*, sig^*)$. Now in **Game-2**, the one-time signature key pair (vk^*, sk^*) is generated in the start of the game. During the game, if $\mathcal{A}$ requests for opening of valid signatures of the form $\Sigma = (\rho, \mathbf{c}, \mathbf{t}, \pi, vk, sig)$, where $vk = vk^*$ then the challenger outputs a random bit and aborts.

It follows that **Game-2** and **Game-1** are indistinguishable. Note that before the challenge phase, vk^* is independent of $\mathcal{A}$'s view, and thus, the probability that vk^* shows up in $\mathcal{A}$'s requests is negligible. Furthermore, after seeing Σ^* , if $\mathcal{A}$ comes up with a valid signature $\Sigma = (\rho, \mathbf{c}, \mathbf{t}, \pi, vk, sig)$ such that $vk = vk^*$, then this implies a forgery against $\mathcal{OTS}$. Therefore, the probability that the challenger aborts in this experiment is negligible and in particular

$$|\text{Adv}_2 - \text{Adv}_1| \leq \text{Adv}_{\mathcal{A}}^{\mathcal{OTS}} + \text{negl}(\lambda),$$

where $\text{Adv}_{\mathcal{A}}^{\mathcal{OTS}}$ is the advantage of $\mathcal{A}$ against strong one-time unforgeability of $\mathcal{OTS}$. In the subsequent

games, we can assume that $\mathcal{A}$ does not request for opening of valid signatures that include vk^* .

Game-3: In this game, we change how $\mathcal{O}_{\text{Join}}$ functions when an identifier $\text{id} \in \{\text{id}'_0, \text{id}'_1\}$ is issued. In the private execution of Join when id is issued, the vector $\mathbf{y}$ is chosen uniformly random in $\mathbb{Z}_{q'}^{m_{\mathbf{B}}}$, instead of being in the form $\mathbf{B}^\top \cdot \mathbf{x} + \mathbf{e} \bmod q'$. In addition, challenger chooses tracing trapdoor $\mathbf{x}$ uniformly random in $\mathbb{Z}_q^n$, instead of faithfully computing $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$ where $\mathbf{z} = \text{usk}_{\text{id}}$. By Lemma 5, the change in the distribution of $\mathbf{x}$ is noticeable with an advantage up to $\mathcal{O}(2^{-\lambda})$. Additionally, the change in the distribution of $\mathbf{y}$ is noticeable up to a negligible advantage, assuming $\text{LWE}_{n,q',D_{\text{LWE}}}$ is hard. This gives

$$|\text{Adv}_3 - \text{Adv}_2| \leq \text{Adv}_{\mathcal{A}}^{\text{LWE}_{n,q',D_{\text{LWE}}}} + \mathcal{O}(2^{-\lambda}),$$

where $\text{Adv}_{\mathcal{A}}^{\text{LWE}_{n,q',D_{\text{LWE}}}}$ is the advantage of $\mathcal{A}$ against $\text{LWE}_{n,q',D_{\text{LWE}}}$.

Game-4: This game is identical to **Game-3**, but we change the way the NIZKAoK π in signatures of id'_0 and id'_1 is created. At the start of the game, the challenger chooses descriptions for $\mathcal{H}_{\text{Sign}}^{(1)}$, $\mathcal{H}_{\text{Sign}}^{(2)}$ as in Proposition 2. When $\mathcal{A}$ requests a signature on behalf of id'_1 or id'_0 (including the challenge signature), the NIZKAoK π is simulated by the NIZK simulator, using the description of $\mathcal{H}_{\text{Sign}}^{(1)}$ and $\mathcal{H}_{\text{Sign}}^{(2)}$. The view of $\mathcal{A}$ can only changed by a negligible quantity compared to **Game-3**, otherwise $\mathcal{A}$ is a distinguisher against the zero-knowledge property of the NIZK argument. In particular, we have that

$$|\text{Adv}_4 - \text{Adv}_3| \leq \text{Adv}_{\mathcal{A}}^{\text{zk}, \mathcal{R}_{\text{Sign}}},$$

where $\text{Adv}_{\mathcal{A}}^{\text{zk}, \mathcal{R}_{\text{Sign}}}$ denotes the advantage of $\mathcal{A}$ against zero-knowledge property of the NIZK argument system for relation $\mathcal{R}_{\text{Sign}}$.

Game-5: This game is identical to **Game-4**, but we change the way $\mathcal{O}_{\text{Claim}}$ responses to queries. At the start of the game, the challenger choosing descriptions for $\mathcal{H}_{\text{Claim}}^{(1)}$, $\mathcal{H}_{\text{Claim}}^{(2)}$ as in Proposition 2. Whenever the adversary queries $\mathcal{O}_{\text{Claim}}$ on (id, M, Σ) such that $\text{id} \in \{\text{id}'_1, \text{id}'_0\}$, the claim χ on the tag $\mathbf{t}$ in Σ is simulated by the NIZK simulator, using the description of $\mathcal{H}_{\text{Claim}}^{(1)}$ and $\mathcal{H}_{\text{Claim}}^{(2)}$. The view of $\mathcal{A}$ can only changed up to a negligible quantity compared to **Game-4**, otherwise $\mathcal{A}$ is a distinguisher against the zero-knowledge property of the NIZK argument. In particular, we have that

$$|\text{Adv}_5 - \text{Adv}_4| \leq \text{Adv}_{\mathcal{A}}^{\text{zk}, \mathcal{R}_{\text{Claim}}},$$

where $\text{Adv}_{\mathcal{A}}^{\text{zk}, \mathcal{R}_{\text{Claim}}}$ denotes the advantage of $\mathcal{A}$ against zero-knowledge property of the NIZK argument system for relation $\mathcal{R}_{\text{Claim}}$.

Game-6: In this game, we change the way the tags $\mathbf{t}$ are computed in the signatures of user id'_0 and id'_1 . Instead of computing $\mathbf{t}$ faithfully, when the adversary $\mathcal{A}$ requests a signature on behalf of id'_0 or id'_1 (including the challenge signature), the challenger simply samples $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_{q'}^{m_{\mathbf{M}}}$ as the tag.

We analyze the change of $\mathcal{A}$'s view. In **Game-5**, the tags included in the signatures from either id'_0 or id'_1 are in the form $\mathbf{M} \cdot \mathbf{x} + \mathbf{e}$, for $\mathbf{e} \leftarrow D_{\text{LWE}}^{m_{\mathbf{M}}}$, $\mathbf{x} \in \mathbb{Z}_q^n$ that is uniformly distributed and $\mathbf{M}$ computed by random

oracle $\mathcal{H}_{\text{LWE}}$. As $\mathcal{A}$ can obtain at most q_{Sign} signatures from either id'_0 or id'_1 , any non-negligible change in $\mathcal{A}$'s view immediately implies a distinguisher for $\text{LWE}_{n,q',D_{\text{LWE}},\mathcal{H}_{\text{LWE}}}$. In particular,

$$|\text{Adv}_6 - \text{Adv}_5| \leq q_{\text{Sign}} \cdot \text{Adv}_{\mathcal{A}}^{\text{LWE}_{n,q',D_{\text{LWE}},\mathcal{H}_{\text{LWE}}}} = q_{\text{Sign}} \cdot \text{Adv}_{\mathcal{A}}^{\text{LWE}_{n,q',D_{\text{LWE}}}},$$

where $\text{Adv}_{\mathcal{A}}^{\text{LWE}_{n,q',D_{\text{LWE}}}}$ denotes the advantage of $\mathcal{A}$ against $\text{LWE}_{n,q',D_{\text{LWE}}}$.

Game-7: In this game, we modify the generation of matrix $\mathbf{B} \in \mathbb{Z}_{q'}^{n \times m_{\mathbf{B}}}$ and change the way $\mathcal{O}_{\text{Reveal}}$ responds to queries. Instead of generating $\mathbf{B}$ along with a $\mathbf{G}$ -trapdoor $\mathbf{S}_{\mathbf{B}}$, we now sample $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_{q'}^{n \times m_{\mathbf{B}}}$. Note that, by Lemma 5, the matrix $\mathbf{B}$ in **Game-6** has a distribution statistically close to uniform. For answering queries to $\mathcal{O}_{\text{Reveal}}$ on input $\text{id} \notin \{\text{id}'_0, \text{id}'_1\}$ (without $\mathbf{S}_{\mathbf{B}}$, which is not available in this game), the challenger simply retrieves $\text{usk}_{\text{id}} = \mathbf{z} \in \{0, 1\}^{m_{\mathbf{B}}}$ and returns $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$. We remark this change to $\mathcal{O}_{\text{Reveal}}$ is syntactical, since $\mathcal{O}_{\text{Reveal}}$ only accepts queries on honest identities and thus challenger knows corresponding secret usk_{id} . In addition, with overwhelming probability, $\mathcal{O}_{\text{Reveal}}$ returns the same $\mathbf{x}$ if challenger uses $\mathbf{S}_{\mathbf{B}}$ to invert the LWE sample $\mathbf{y}$. Therefore,

$$|\text{Adv}_7 - \text{Adv}_6| \leq \text{negl}(\lambda).$$

Game-8: In this game, we program the random oracle $\mathcal{H}_{\text{GPV}}$ as follows: at the start of the game, the challenger choose a random oracle $\mathcal{H}'_{\text{GPV}}$ that maps input in $\{0, 1\}^l$ to output in the randomness space of distribution $\mathcal{D}_{\sigma_{\text{GPV}}}^{m_{\mathbf{B}}}$; the random oracle $\mathcal{H}_{\text{GPV}}$, when queried on input $vk \in \{0, 1\}^l$, returns $\mathbf{v} = \mathbf{B} \cdot \mathbf{e} \bmod q'$, where $\mathbf{e}$ is sampled from $\mathcal{D}_{\sigma_{\text{GPV}}}^{m_{\mathbf{B}}}$ using randomness $\mathcal{H}'_{\text{GPV}}(vk)$. The challenger also records $(vk, \mathbf{v}, \mathbf{e})$ to answer queries to $\mathcal{O}_{\text{Open}}$ that involves vk . Following the proof of [49, Theorem 2], we have that

$$|\text{Adv}_8 - \text{Adv}_7| \leq Q^2 \cdot 2^{-\Omega(n)},$$

assuming $\mathcal{H}_{\text{GPV}}$ is accessed at most Q times. Observe that challenger invokes $\mathcal{H}_{\text{GPV}}$ whenever $\mathcal{A}$ queries $\mathcal{O}_{\text{Sign}}, \mathcal{O}_{\text{Open}}$ and $\mathcal{H}_{\text{GPV}}$ directly. Therefore $Q \leq q_{\text{Sign}} + q_{\text{Open}} + q_{\mathcal{H}_{\text{GPV}}}$ and we have

$$|\text{Adv}_8 - \text{Adv}_7| \leq (q_{\text{Sign}} + q_{\text{Open}} + q_{\mathcal{H}_{\text{GPV}}})^2 \cdot 2^{-\Omega(m_{\mathbf{B}})}.$$

Game-9: In this game, we change how the ciphertext $\mathbf{c}^*$ in the challenge signature Σ^* is generated. Note that in **Game-8**, we have

$$\mathbf{c}^* = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^{*\top} \end{pmatrix} \cdot \mathbf{r}^* + \mathbf{e}_{\mathbf{c}}^* + \begin{pmatrix} \mathbf{0}^n \\ \lceil q'/2(N+1) \rceil \cdot \text{id}_b^* \end{pmatrix} \bmod q' = \begin{pmatrix} \mathbf{c}_1^* \\ c_1^* \end{pmatrix} \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}} \times \mathbb{Z}_{q'};$$

where $\mathbf{v}^* = \mathcal{H}_{\text{GPV}}(vk^*)$, $\mathbf{r}^* \xleftarrow{\$} \mathbb{Z}_{q'}^n$ and $\mathbf{e}_{\mathbf{c}}^* \leftarrow \mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}^{m_{\mathbf{B}}+1}$. Now the challenger generates $\mathbf{c}^*$ as follows:

1. Sample $\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_{q'}^n$;
2. Sample $\mathbf{e}' \leftarrow \mathcal{D}_{\alpha \cdot q'}^{m_{\mathbf{B}}}$;

3. Apply the algorithm ReRand in [49, Lemma 7] on input $\mathbf{V} = (\mathbf{I}_n | \mathbf{v}^*)$, $\mathbf{b} = \mathbf{B}^\top \cdot \mathbf{r}' \bmod q'$, $\mathbf{z} = \mathbf{e}'$, $r = \alpha \cdot q'$ and $\sigma = \alpha_{\text{GPV}}/2\alpha$ to generate $\mathbf{c}' = (c'_1, c'_1) \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}} \times \mathbb{Z}_{q'}$. This algorithm ensures that $\mathbf{c}'$ has the form

$$\mathbf{V}^\top \cdot \mathbf{b} + \mathbf{e} = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^{*\top} \end{pmatrix} \cdot \mathbf{r}' + \mathbf{e},$$

where $\mathbf{e} \in \mathbb{Z}^{m_{\mathbf{B}}+1}$ is distributed within a statistical distance at most $2^{-\Omega(n)}$ to $\mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}^{m_{\mathbf{B}}}$;

4. Output $\mathbf{c}^* = \mathbf{c}' + (\mathbf{0}, \lceil q'/2(N+1) \rceil \cdot \text{id}_b^*) \bmod q'$.

It follows that

$$|\text{Adv}_9 - \text{Adv}_8| \leq 2^{-\Omega(n)}.$$

Game-10: In this game, we change how the ciphertext $\mathbf{c}^*$ in the challenge signature Σ^* is generated. The process is identical to **Game-9** except that the vector $\mathbf{b} = \mathbf{B}^\top \cdot \mathbf{r}' \bmod q' \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}}$ is now replaced by a uniformly random $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_{q'}^{m_{\mathbf{B}}}$. It follows that the vector $\mathbf{c}' = (c'_1, c'_1) \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}} \times \mathbb{Z}_{q'}$ output by ReRand has the form

$$\mathbf{V}^\top \cdot \mathbf{b} + \mathbf{e} = \begin{pmatrix} \mathbf{I}_n \\ \mathbf{v}^{*\top} \end{pmatrix} \cdot \mathbf{b} + \mathbf{e} \bmod q',$$

where $\mathbf{e} \in \mathbb{Z}^{m_{\mathbf{B}}+1}$ is distributed within a statistical distance at most $2^{-\Omega(n)}$ to $\mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}^{m_{\mathbf{B}}}$. By a similar argument in the proof of [49, Theorem 2], we have that **Game-10** and **Game-9** are indistinguishable assuming $\text{LWE}_{n, q', \mathcal{D}_{\alpha \cdot q'}, \mathcal{H}'_{\text{GPV}}}$ is hard. Thus,

$$|\text{Adv}_{10} - \text{Adv}_9| \leq \text{Adv}_{\mathcal{A}}^{\text{LWE}_{n, q', \mathcal{D}_{\alpha \cdot q'}}},$$

where $\text{Adv}_{\mathcal{A}}^{\text{LWE}_{n, q', \mathcal{D}_{\alpha \cdot q'}}}$ denotes the advantage of $\mathcal{A}$ against $\text{LWE}_{n, q', \mathcal{D}_{\alpha \cdot q'}}$.

Game-11: In this game, we change how the ciphertext $\mathbf{c}^*$ in the challenge signature Σ^* is generated as follows: sample $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_{q'}^{m_{\mathbf{B}}}$, sample $\mathbf{e} \leftarrow \mathcal{D}_{\alpha_{\text{GPV}} \cdot q'}^{m_{\mathbf{B}}}$, compute

$$\mathbf{c}' = \begin{pmatrix} \mathbf{I}_n \\ \mathbf{v}^{*\top} \end{pmatrix} \cdot \mathbf{b} + \mathbf{e} \bmod q' \in \mathbb{Z}_{q'}^{m_{\mathbf{B}}+1},$$

and output $\mathbf{c}^* = \mathbf{c}' + (\mathbf{0}, \lceil q'/2(N+1) \rceil \cdot \text{id}_b^*) \bmod q'$. By a similar argument in the proof of [49, Theorem 2], **Game-11** and **Game-10** are distinguishable up to an advantage of order $2^{-\Omega(n)}$ and in addition, $\mathbf{c}^*$ has a distribution statistically close to uniform over $\mathbb{Z}_{q'}^{m_{\mathbf{B}}+1}$.

Finally, $\mathcal{A}$ can only win **Game-11** with probability negligibly close to $1/2$. To see this, note that to $\mathcal{A}$'s view, the certificates of id_0^* and id_1^* reveal nothing about the respective tracing trapdoors. Any claim χ on behalf of id_0^* or id_1^* is simulated. In the signatures of id_0^* or id_1^* , the proof π is simulated and the tag $\mathbf{t}$ is uniformly distributed. As such, oracle outputs involving id_0^* or id_1^* reveals nothing about their tracing keys

and their secrets. Furthermore, every component of the challenge signature is generated independently with challenger's random bit b and the ciphertext $\mathbf{c}^*$ statistically hides id_b^* . It follows that the traceable scheme in Section 3.2 is CCA-anonymous. $\square$

Lemma 13 (Traceability). *The TS scheme in Section 3.2 is traceable in QROM, assuming that: (i) the signature scheme of [44] is unforgeable under chosen-message attacks; (ii) the NIZK argument system is simulation-sound online-extractable; and (iii) the $\text{SIS}_{n, m_{\mathbf{F}}+1, q, 1}^\infty$ problem is hard.*

PROOF. Consider the experiment $\text{Exp}_{\mathcal{T}, \mathcal{A}}^{\text{trace}}(\lambda)$ presented in Figure A.1. A challenger $\mathcal{C}$ sets up the public key and provides the adversary $\mathcal{A}$ with input and oracle access (by interacting with $\mathcal{A}$, acting as group manager). In addition, it programs the random oracles $\mathcal{H}_{\text{Sign}}^{(1)}, \mathcal{H}_{\text{Sign}}^{(2)}$ using Proposition 2. Since the simulation can be done perfectly, the view of any PPT quantum adversary $\mathcal{A}$ remains unchanged.

Whenever $\mathcal{A}$ outputs a valid signature (M^*, Σ^*) , the challenger $\mathcal{C}$ parses $\Sigma^* = (vk^*, \rho^*, \mathbf{c}^*, \mathbf{t}^*, \pi^*, sig^*)$ as in (2). Then it reconstructs the statement of relation $\mathcal{R}_{\text{Sign}}$ in Definition 8 and invokes the NIZK extractor on the NIZKAoK π^* with the oracle description of $\mathcal{H}_{\text{Sign}}^{(1)}, \mathcal{H}_{\text{Sign}}^{(2)}$. Let $\zeta = (\text{id}^*, \mathbf{z}^*, \mathbf{x}^*, \mathbf{e}^*, \text{cert}_{\text{id}}^* = (\mathbf{y}^*, \mathbf{v}_1^*, \mathbf{v}_2^*), \mathbf{r}^*, \mathbf{e}_{\mathbf{c}}^*, \mathbf{e}_{\mathbf{t}}^*)$ be the output of NIZK extractor. As the NIZK system is online-extractable in QROM, ζ is a valid witness of $\mathcal{R}_{\text{Sign}}$ with overwhelming probability. In particular:

- $\mathbf{x}^* = \mathbf{F} \cdot \mathbf{z}^* \bmod q'$;
- $(\mathbf{A} \mid \text{id}^* \cdot \mathbf{G}_n + \mathbf{A}') \cdot (\mathbf{v}_1^*, \mathbf{v}_2^*) = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{y}^*) \bmod q$;
- $\mathbf{y}^* = \mathbf{B}^\top \cdot \mathbf{x}^* + \mathbf{e}^* \bmod q'$;
- $\mathbf{c}^* = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r}^* + \mathbf{e}_{\mathbf{c}}^* + \begin{pmatrix} \mathbf{0}^{m_{\mathbf{B}}} \\ \lceil q/(2(N+1)) \rceil \cdot \text{id}^* \end{pmatrix} \bmod q'$;
- $\mathbf{t}^* = \mathbf{M}^* \cdot \mathbf{x}^* + \mathbf{e}_{\mathbf{t}}^* \bmod q'$;

where $\mathbf{M}^* = \mathcal{H}_{\text{PRF}}(\rho^*)$.

Note that with overwhelming probability, $(\text{id}^*, \mathbf{v}_1^*, \mathbf{v}_2^*, \mathbf{y}^*)$ belongs to some user which joined the group via either $\mathcal{O}_{\text{pJoin}}$ or $\mathcal{O}_{\text{aJoin}}$. If this is not the case, $\mathcal{C}$ immediately obtains a forgery of JRS signature scheme. Furthermore, the equalities

$$\mathbf{y}^* = \mathbf{B}^\top \cdot \mathbf{x}^* + \mathbf{e}^* \bmod q'$$

and

$$\mathbf{t}^* = \mathbf{M}^* \cdot \mathbf{x}^* + \mathbf{e}_{\mathbf{t}}^* \bmod q'$$

imply that the output of `Reveal` on id^* is $\mathbf{x}^*$, and Σ^* is traced to id^* when running `Trace` on tracing key $\mathbf{x}^*$. In addition, the equality

$$\mathbf{c}^* = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r}^* + \mathbf{e}_{\mathbf{c}}^* + \begin{pmatrix} \mathbf{0}^{m_{\mathbf{B}}} \\ \lceil q/(2(N+1)) \rceil \cdot \text{id}^* \end{pmatrix} \bmod q'$$

implies that Σ^* is opened to id^* . Either way, $\mathcal{A}$ wins if and only if $\text{id}^* \in \text{Honest}$, i.e. id^* was issued in an execution of $\mathcal{O}_{\text{pJoin}}$ and was not queried to $\mathcal{O}_{\text{Corrupt}}$. We now show that in this case $\mathcal{C}$ can solve $\text{SIS}_{n,m+1,q,1}^\infty$, using $\mathcal{A}$ as subroutine. The details are as follows.

Setup: $\mathcal{C}$ receives an SIS instance $\mathbf{A}_{\text{SIS}} \xleftarrow{\$} \mathbb{Z}_{q'}^{n \times (m_{\mathbf{F}}+1)}$. It parses $\mathbf{A}_{\text{SIS}} = (\mathbf{F}' \mid -\mathbf{x}')$ in $\mathbb{Z}_{q'}^{n \times m_{\mathbf{F}}} \times \mathbb{Z}_{q'}^n$. Next, $\mathcal{C}$ sets $\mathbf{F} = \mathbf{F}'$. For the other components in group public key gpk , group manager secret key gsk and opening authority key osk , $\mathcal{C}$ sets them up faithfully following system's description in Section 3.2. Note that, the distribution of gpk in this case is identical to that of the real system. In addition, let q_{pJoin} be the maximum number of queries to $\mathcal{O}_{\text{pJoin}}$. The challenger $\mathcal{C}$ guesses uniformly at random an index $j \in \{1, \dots, q_{\text{pJoin}}\}$.

Handling Queries: The queries are handled exactly like in the real experiment $\text{Exp}_{\mathcal{T}, \mathcal{S}, \mathcal{A}}^{\text{trace}}(\lambda)$ (see Figure A.1) except for the j -th query of $\mathcal{A}$ to $\mathcal{O}_{\text{pJoin}}$. Instead of simulating $\mathcal{O}_{\text{pJoin}}$ in private faithfully, in the j -th query, $\mathcal{C}$ retrieves $\mathbf{x}'$ from **Setup** phase, samples $\mathbf{e}' \leftarrow D_{\text{LWE}}^{m_{\mathbf{B}}}$, computes $\mathbf{y}' = \mathbf{B}^\top \cdot \mathbf{x}' + \mathbf{e}' \bmod q'$ and certifies $\mathbf{y}'$ as group manager. After **Join** terminates, $\mathcal{C}$ stores the relevant information in **info**, but set the corresponding secret usk of user with issued identifier id' to be $\perp$. We remark that, the distribution of $\mathbf{x}'$ in this case is uniform. In comparison, if $\mathcal{C}$ simulates **Join** faithfully by sampling $\mathbf{z} \xleftarrow{\$} \{0, 1\}^{m_{\mathbf{F}}}$ and setting the user's tracing key as $\mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q'$, then the distribution of $\mathbf{x}$ is statistically close to uniform (by Lemma 5).

If $\mathcal{A}$ queries id' to $\mathcal{O}_{\text{Corrupt}}$ then $\mathcal{C}$ aborts. If $\mathcal{A}$ queries id' to $\mathcal{O}_{\text{Sign}}$, it generates the components in the signature Σ faithfully except for the NIZK proof π , which is now simulated by the NIZK simulator. In $\mathcal{A}$'s view, this change is noticeable up to a negligible quantity, since the NIZK is zero-knowledge.

It follows that $\mathcal{C}$ successfully simulates the view of $\mathcal{A}$ in the real experiment $\text{Exp}_{\mathcal{T}, \mathcal{S}, \mathcal{A}}^{\text{trace}}(\lambda)$, and $\mathcal{A}$ can only notice with a negligible advantage.

Exploiting Forgery: After $\mathcal{A}$ submits its forgery (M^*, Σ^*) , the challenger runs the NIZK extractor to obtain a witness $(\text{id}^*, \mathbf{z}^*, \mathbf{x}^*, \mathbf{e}^*, \text{cert}_{\text{id}}^* = (\mathbf{y}^*, \mathbf{v}_1^*, \mathbf{v}_2^*), \mathbf{r}^*, \mathbf{e}_{\mathbf{c}}^*, \mathbf{e}_{\mathbf{t}}^*)$. As the NIZK system is simulation-sound online-extractable, the extracted witness is valid with overwhelming probability.

Now $\mathcal{C}$ checks if $(\text{id}', \mathbf{y}') = (\text{id}^*, \mathbf{y}^*)$. If this is not the case then $\mathcal{C}$ aborts. As assumed before, $(\text{id}^*, \mathbf{y}^*)$ was involved prior in some query of $\mathcal{O}_{\text{pJoin}}$. Since the guess of $\mathcal{C}$ is independent of $\mathcal{A}$'s view, $\mathcal{C}$ succeeds with probability at least $1/q_{\text{pJoin}}$. Finally, $\mathcal{C}$ recovers $\mathbf{z}^*$ and outputs $(\mathbf{z}^*, 1) \in \{0, 1\}^{m_{\mathbf{F}}+1}$ as the SIS solution to $\mathbf{A}_{\text{SIS}}$. It is easy to see that $(\mathbf{z}^*, 1)$ is a valid solution to the SIS instance $\mathbf{A}_{\text{SIS}}$. $\square$

We note that the unforgeability of the signature scheme by Jeudy *et al.* [44] is based on SIS assumptions.

Lemma 14 (Non-frameability). *The TS scheme in Section 3.2 is non-frameable in QROM, assuming that: (i) the underlying NIZK argument is simulation-sound online-extractable; and (ii) the $\text{SIS}_{n,m_{\mathbf{F}},q,1}^\infty$ problem is hard.*

PROOF. Consider the experiment $\text{Exp}_{\mathcal{T}\mathcal{S}, \mathcal{A}}^{\text{frame}}(\lambda)$ presented in Figure A.2. A challenger $\mathcal{C}$ sets up the public key and provides the adversary $\mathcal{A}$ with necessary inputs and oracle access. In addition, it programs the random oracles $\mathcal{H}_{\text{Sign}}^{(1)}$, $\mathcal{H}_{\text{Sign}}^{(2)}$, $\mathcal{H}_{\text{Claim}}^{(1)}$, $\mathcal{H}_{\text{Claim}}^{(2)}$ using Proposition 2. As simulation of oracles is perfect, the view of $\mathcal{A}$ remains unchanged.

Whenever $\mathcal{A}$ outputs a valid signature (M^*, Σ^*) , the challenger $\mathcal{C}$ parses $\Sigma^* = (vk^*, \rho^*, \mathbf{c}^*, \mathbf{t}^*, \pi^*, sig^*)$ as in (2). Then it reconstructs the statement of relation $\mathcal{R}_{\text{Sign}}$ in Definition 8 and invokes the NIZK extractor on the NIZKAoK π^* with the oracle description of $\mathcal{H}_{\text{Sign}}^{(1)}$, $\mathcal{H}_{\text{Sign}}^{(2)}$. Let $\zeta = (\text{id}^*, \mathbf{z}^*, \mathbf{x}^*, \mathbf{e}^*, \text{cert}_{\text{id}}^* = (\mathbf{y}^*, \mathbf{v}_1^*, \mathbf{v}_2^*), \mathbf{r}^*, \mathbf{e}_{\mathbf{c}}^*, \mathbf{e}_{\mathbf{t}}^*))$ be the output of NIZK extractor. As the NIZK system is online-extractable in QROM, ζ is a valid witness of $\mathcal{R}_{\text{Sign}}$ with overwhelming probability. In particular:

- $\mathbf{x}^* = \mathbf{F} \cdot \mathbf{z}^* \bmod q'$;
- $(\mathbf{A} \mid \text{id}^* \cdot \mathbf{G}_n + \mathbf{A}') \cdot (\mathbf{v}_1^*, \mathbf{v}_2^*) = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{y}^*) \bmod q$;
- $\mathbf{y}^* = \mathbf{B}^\top \cdot \mathbf{x}^* + \mathbf{e}^* \bmod q'$;
- $\mathbf{c}^* = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r}^* + \mathbf{e}_{\mathbf{c}}^* + \begin{pmatrix} \mathbf{0}^{m_{\mathbf{B}}} \\ \lceil q/(2(N+1)) \rceil \cdot \text{id}^* \end{pmatrix} \bmod q'$;
- $\mathbf{t}^* = \mathbf{M}^* \cdot \mathbf{x}^* + \mathbf{e}_{\mathbf{t}}^* \bmod q'$;

where $\mathbf{M}^* = \mathcal{H}_{\text{PRF}}(\rho^*)$. Note that the above imply that the signature is opened to a user with identifier id^* , or can be traced by $\mathbf{x}^*$, which is obtained by running Reveal on input id^* . Therefore, if $\mathcal{A}$ wins by passing the check in line 5 of Figure A.2, it must be the case that $(\text{id}^*, \mathbf{v}_1^*, \mathbf{v}_2^*, \mathbf{y}^*)$ was issued in an execution of $\mathcal{O}_{\text{pJoin}}$ and id^* was not queried to $\mathcal{O}_{\text{Corrupt}}$. In such case, similar to the proof of Lemma 13 we can construct an algorithm solving $\text{SIS}_{n, m_{\mathbf{F}}+1, q', 1}^\infty$, using $\mathcal{A}$ as a subroutine.

In the case that $\mathcal{A}$ forges a claim χ^* on a signature (M^*, Σ^*) created by some honest user with identifier id^* , $\mathcal{C}$ can solve $\text{SIS}_{n, m_{\mathbf{F}}+1, q', 1}^\infty$ using $\mathcal{A}$ as a subroutine. The reduction is nearly identical: $\mathcal{C}$ guesses the targeted id^* and embeds the SIS instance; when exploiting the forgery, it runs the extractor on the forged claim χ^* to obtain an SIS solution. $\square$

Combining the results of Lemma 12, Lemma 13 and Lemma 14, Theorem 11 then follows.

References

- [1] D. Chaum, E. van Heyst, Group signatures, in: EUROCRYPT '91, Vol. 547 of LNCS, Springer, 1991, pp. 257–265.
- [2] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, A practical and provably secure coalition-resistant group signature scheme, in: CRYPTO 2000, Vol. 1880 of LNCS, Springer, 2000, pp. 255–270.

- [3] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: CRYPTO 2004, Vol. 3152 of LNCS, Springer, 2004, pp. 41–55.
- [4] A. Kiayias, M. Yung, Group signatures with efficient concurrent join, in: EUROCRYPT 2005, Vol. 3494 of LNCS, Springer, 2005, pp. 198–214.
- [5] X. Boyen, B. Waters, Compact group signatures without random oracles, in: EUROCRYPT 2006, Vol. 4004 of LNCS, Springer, 2006, pp. 427–444.
- [6] J. Groth, Fully anonymous group signatures without random oracles, in: ASIACRYPT 2007, Vol. 4833 of LNCS, Springer, 2007, pp. 164–180.
- [7] S. D. Gordon, J. Katz, V. Vaikuntanathan, A group signature scheme from lattice assumptions, in: ASIACRYPT 2010, Vol. 6477 of LNCS, Springer, 2010, pp. 395–412.
- [8] F. Laguillaumie, A. Langlois, B. Libert, D. Stehlé, Lattice-based group signatures with logarithmic signature size, in: ASIACRYPT 2013, Vol. 8270 of LNCS, Springer, 2013, pp. 41–61.
- [9] B. Libert, S. Ling, K. Nguyen, H. Wang, Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors, in: EUROCRYPT 2016, Vol. 9666 of LNCS, Springer, 2016, pp. 1–31.
- [10] S. Ling, K. Nguyen, H. Wang, Y. Xu, Constant-size group signatures from lattices, in: PKC 2018, Vol. 10770 of LNCS, Springer, 2018, pp. 58–88.
- [11] R. del Pino, V. Lyubashevsky, G. Seiler, Lattice-based group signatures and zero-knowledge proofs of automorphism stability, in: CCS 2018, ACM, 2018, pp. 574–591.
- [12] V. Lyubashevsky, N. K. Nguyen, M. Plançon, G. Seiler, Shorter lattice-based group signatures via "almost free" encryption and other optimizations, in: ASIACRYPT 2021, Vol. 13093 of LNCS, Springer, 2021, pp. 218–248.
- [13] V. Lyubashevsky, N. K. Nguyen, M. Plançon, Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general, in: CRYPTO 2022, Vol. 13508 of LNCS, Springer, 2022, pp. 71–101.
- [14] W. Beullens, S. Dobson, S. Katsumata, Y. Lai, F. Pintore, Group signatures and more from isogenies and lattices: generic, simple, and efficient, Des. Codes Cryptogr. 91 (6) (2023) 2141–2200.
- [15] R. E. Bansarkhani, R. Misoczki, G-merkle: A hash-based group signature scheme from standard assumptions, in: PQCrypto 2018, Vol. 10786 of LNCS, Springer, 2018, pp. 441–463.
- [16] M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, H. Wang, A provably secure group signature scheme from code-based assumptions, in: ASIACRYPT 2015, Vol. 9452 of LNCS, Springer, 2015, pp. 260–285.

- [17] K. Nguyen, H. Tang, H. Wang, N. Zeng, New code-based privacy-preserving cryptographic constructions, in: ASIACRYPT 2019, Vol. 11922 of LNCS, Springer, 2019, pp. 25–55.
- [18] O. Blazy, P. Gaborit, D. T. Mac, A rank metric code-based group signature scheme, in: CBCrypto 2021, Vol. 13150 of LNCS, Springer, 2021, pp. 1–21.
- [19] Y. Ouyang, D. Tang, Y. Xu, Code-based zero-knowledge from vole-in-the-head and their applications: Simpler, faster, and smaller, in: K. Chung, Y. Sasaki (Eds.), ASIACRYPT 2024, Vol. 15488 of LNCS, Springer, 2024, pp. 436–470.
- [20] D. Boneh, H. Shacham, Group signatures with verifier-local revocation, in: CCS 2004, ACM, 2004, pp. 168–177.
- [21] M. Kohlweiss, I. Miers, Accountable metadata-hiding escrow: A group signature case study, *Proc. Priv. Enhancing Technol.* 2015 (2) (2015) 206–221.
- [22] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, K. Omote, Group signatures with message-dependent opening, in: Pairing-Based Cryptography 2012, Vol. 7708 of LNCS, Springer, 2012, pp. 270–294.
- [23] A. El Kaafarani, E. Ghadafi, D. Khader, Decentralized traceable attribute-based signatures, in: CT-RSA 2014, Vol. 8366 of LNCS, Springer, 2014, pp. 327–348.
- [24] B. Libert, K. Nguyen, T. Peters, M. Yung, Bifurcated signatures: Folding the accountability vs. anonymity dilemma into a single private signing scheme, in: EUROCRYPT 2021, Vol. 12698 of LNCS, Springer, 2021, pp. 521–552.
- [25] K. Nguyen, F. Guo, W. Susilo, G. Yang, Multimodal private signatures, in: CRYPTO 2022, Vol. 13508 of LNCS, Springer, 2022, pp. 792–822.
- [26] B. Libert, F. Mouhartem, K. Nguyen, A lattice-based group signature scheme with message-dependent opening, in: ACNS 2016, Vol. 9696 of LNCS, Springer, 2016, pp. 137–155.
- [27] S. Cheng, K. Nguyen, H. Wang, Policy-based signature scheme from lattices, *Des. Codes Cryptogr.* 81 (1) (2016) 43–74.
- [28] S. Ling, K. Nguyen, A. Roux-Langlois, H. Wang, A lattice-based group signature scheme with verifier-local revocation, *Theor. Comput. Sci.* 730 (2018) 1–20.
- [29] S. Ling, K. Nguyen, H. Wang, Y. Xu, Accountable tracing signatures from lattices, in: CT-RSA 2019, Vol. 11405 of LNCS, Springer, 2019, pp. 556–576.

- [30] A. Kiayias, Y. Tsiounis, M. Yung, Traceable signatures, in: EUROCRYPT 2004, 2004, pp. 571–589.
- [31] A. Ishida, Y. Sakai, K. Emura, G. Hanaoka, K. Tanaka, Fully anonymous group signature with verifier-local revocation, in: SCN 2018, Vol. 11035 of LNCS, Springer, 2018, pp. 23–42.
- [32] Y. Zhang, X. Liu, Y. Hu, Q. Zhang, H. Jia, Lattice-based group signatures with verifier-local revocation: Achieving shorter key-sizes and explicit traceability with ease, in: CANS 2019, Vol. 11829 of LNCS, Springer, 2019, pp. 120–140.
- [33] S. Chen, J. Chen, A. Miyaji, K. Chen, Constant-size group signatures with message-dependent opening from lattices, in: ProvSec 2023, Vol. 14217 of LNCS, Springer, 2023, pp. 166–185.
- [34] B. Libert, M. Yung, Efficient traceable signatures in the standard model, in: Pairing-Based Cryptography, 2009, pp. 187–205.
- [35] O. Blazy, D. Pointcheval, Traceable signature with stepping capabilities, in: Cryptography and Security: From Theory to Applications, Vol. 6805 of LNCS, Springer, 2012, pp. 108–131.
- [36] T. Preethi, B. Amberker, Traceable signatures using lattices, Int. Arab J. Inf. Technol. 17 (6) (2020) 965–975.
- [37] J. Camenisch, J. Groth, Group signatures: Better efficiency and new theoretical aspects, in: SCN 2004, Vol. 3352 of LNCS, Springer, 2004, pp. 120–133.
- [38] M. Bellare, H. Shi, C. Zhang, Foundations of group signatures: The case of dynamic groups, in: CT-RSA 2005, Vol. 3376 of LNCS, Springer, 2005, pp. 136–153.
- [39] S. G. Choi, K. Park, M. Yung, Short traceable signatures based on bilinear pairings, in: IWSEC 2006, 2006, pp. 88–103.
- [40] S. S. M. Chow, Real traceable signatures, in: SAC 2009, 2009, pp. 92–107.
- [41] Y. Dodis, A. Yampolskiy, A verifiable random function with short proofs and keys, in: S. Vaudenay (Ed.), PKC 2005, Vol. 3386 of LNCS, Springer, 2005, pp. 416–431.
- [42] A. Fiat, A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, in: CRYPTO 1986, Vol. 263 of LNCS, Springer, 1986, pp. 186–194.
- [43] J. Groth, A. Sahai, Efficient non-interactive proof systems for bilinear groups, in: EUROCRYPT 2008, Vol. 4965 of LNCS, Springer, 2008, pp. 415–432.
- [44] C. Jeudy, A. Roux-Langlois, O. Sanders, Lattice signature with efficient protocols, application to anonymous credentials, in: CRYPTO 2023, Vol. 14082 of LNCS, Springer, 2023, pp. 351–383.

- [45] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in: STOC 2008, ACM, 2008, pp. 197–206.
- [46] R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, W. Whyte, Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications, in: CRYPTO 2019, Vol. 11692 of LNCS, Springer, 2019, pp. 147–175.
- [47] D. Unruh, Non-interactive zero-knowledge proofs in the quantum random oracle model, in: EUROCRYPT 2015, Vol. 9057 of LNCS, Springer, 2015, pp. 755–784.
- [48] H. Feng, J. Liu, Q. Wu, Secure stern signatures in quantum random oracle model, in: ISC 2019, Vol. 11723 of LNCS, Springer, 2019, pp. 425–444.
- [49] S. Katsumata, S. Yamada, T. Yamakawa, Tighter security proofs for GPV-IBE in the quantum random oracle model, in: ASIACRYPT 2018, Vol. 11273 of LNCS, Springer, 2018, pp. 253–282.
- [50] M. Ajtai, Generating hard instances of lattice problems (extended abstract), in: STOC 1996, ACM, 1996, pp. 99–108.
- [51] R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, in: EUROCRYPT 2004, Vol. 3027 of LNCS, Springer, 2004, pp. 207–222.
- [52] J. Stern, A new paradigm for public key identification, IEEE Trans. Inf. Theory 42 (6) (1996) 1757–1768.
- [53] M. Abe, S. S. M. Chow, K. Haralambiev, M. Ohkubo, Double-trapdoor anonymous tags for traceable signatures, in: ACNS 2011, Vol. 6715 of LNCS, 2011, pp. 183–200.
- [54] S. Park, A. Sealfon, It wasn’t me! repudiability and claimability of ring signatures, in: CRYPTO 2019, Vol. 110694 of LNCS, Springer, 2019, pp. 159–190.
- [55] S. Katsumata, S. Yamada, T. Yamakawa, Tighter security proofs for GPV-IBE in the quantum random oracle model, Journal of Cryptology 34 (1) (2021) 5.
- [56] A. Langlois, S. Ling, K. Nguyen, H. Wang, Lattice-based group signature scheme with verifier-local revocation, in: PKC 2014, Vol. 8383 of LNCS, Springer, 2014, pp. 345–361.
- [57] S. Goldwasser, Y. T. Kalai, C. Peikert, V. Vaikuntanathan, Robustness of the learning with errors assumption, in: ICS 2010, 2010, pp. 230–240.
- [58] W. Banaszczyk, New bounds in some transference theorems in the geometry of numbers, Mathematische Annalen 296 (1993) 625–635.

- [59] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, in: STOC 2005, ACM, 2005, pp. 84–93.
- [60] D. Micciancio, C. Peikert, Trapdoors for lattices: Simpler, tighter, faster, smaller, in: EUROCRYPT 2012, Vol. 7237 of LNCS, Springer, 2012, pp. 700–718.
- [61] S. Ling, K. Nguyen, D. Stehlé, H. Wang, Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications, in: PKC 2013, Vol. 7778 of LNCS, Springer, 2013, pp. 107–124.
- [62] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, C. Peikert, More efficient commitments from structured lattice assumptions, in: SCN 2018, Vol. 11035 of LNCS, Springer, 2018, pp. 368–385.
- [63] P. Mohassel, One-time signatures and chameleon hash functions, in: SAC 2010, Vol. 6544 of LNCS, Springer, 2010, pp. 302–319.
- [64] M. Zhandry, Secure identity-based encryption in the quantum random oracle model, in: CRYPTO 2012, Vol. 7417 of LNCS, Springer, 2012, pp. 758–775.
- [65] A. Kawachi, K. Tanaka, K. Xagawa, Concurrently secure identification schemes based on the worst-case hardness of lattice problems, in: ASIACRYPT 2008, Vol. 5350 of LNCS, Springer, 2008, pp. 372–389.
- [66] N. Tran, K. Nguyen, D. Liu, J. Pieprzyk, W. Susilo, Improved multimodal private signatures from lattices, in: ACISP 2024, 2024, pp. 3–23.

Appendix A. Security Requirement for Traceable Signatures

We follow the BP model [35]. The model allows an adversary access to numerous oracles. Whenever some of the oracles are invoked, they make changes to databases that are empty from the start. They are defined as below:

- **info**: contains information obtainable from the system point of view.
- **Control**: contains identities of users controlled by adversary.
- **Honest**: contains identities of users that adversary does not know the secrets.
- **SIGN**: contains signatures created by honest users.
- **REV**: contains revealed tracing trapdoors.
- **CLAIM**: contains claims created by honest users.

The oracles are nearly identical to those described in [34, 35]. First, the adversary can introduce users to the group passively (only observing the communication) or actively (playing the role of a user). These are allowed via access to the following oracles:

- $\mathcal{O}_{\text{pJoin}}$: adds new honest users to the group. It simulates `Join` to obtain $(\text{id}, \text{usk}_{\text{id}}, \text{cert}_{\text{id}})$ and $\text{transcript}_{\text{id}}$, then adds id to `Honest` and updates `info` by adding an entry of the form $(\text{id}, \text{usk}_{\text{id}}, \text{cert}_{\text{id}}, \text{transcript}_{\text{id}})$.
- $\mathcal{O}_{\text{aJoin}}$: adds controlled users to the group. The oracle, acting as $\mathcal{GM}$, interacts with the adversary (the controlled user) via `Join`. If `Join` terminates, $\mathcal{O}_{\text{aJoin}}$ adds id to `Control` and updates `info` by adding an entry of the form $(\text{id}, \perp, \text{cert}_{\text{id}}, \text{transcript}_{\text{id}})$.

In framing attacks, adversary can corrupt group manager $\mathcal{GM}$ (obtaining gsk). In such case, it can run $\mathcal{O}_{\text{aJoin}}$ by itself to introduce new controlled users to the group.

After users are added, the adversary can access the following oracles:

- $\mathcal{O}_{\text{Corrupt}}$: takes as input an identifier id . The oracle returns $\perp$ if $\text{id} \notin \text{Honest}$. Else, it returns usk_{id} in the entry of `info` containing id . Finally, it removes this entry from `Honest` and adds it to `Control`.
- $\mathcal{O}_{\text{Sign}}$: receives query on an identifier id and message M , $\mathcal{O}_{\text{Sign}}$ if $\text{id} \in \text{Honest}$. It returns $\perp$ if such id does not exist. Else, it retrieves $(\text{usk}_{\text{id}}, \text{cert}_{\text{id}})$ from the entry containing id in `info`, returns $\Sigma \leftarrow \text{Sign}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), M)$ and sets $\text{SIGN} \leftarrow \text{SIGN} \parallel (\text{id}, M, \Sigma)$.
- $\mathcal{O}_{\text{Reveal}}$: receives query on a user identifier id , $\mathcal{O}_{\text{Reveal}}$ checks if $\text{id} \in \text{Honest}$. It returns $\perp$ if such id does not exist. Else, it returns the output of $\text{Reveal}(\text{gpk}, \text{osk}, \text{id})$ and adds id to `REV`.
- $\mathcal{O}_{\text{Claim}}$: receives query on an identifier id , a message M and a signature Σ , this oracle first checks whether $\text{id} \in \text{Honest}$ and whether there is an entry (id, M, Σ) in `SIGN`. If one of these conditions does not hold, it returns $\perp$. Else, the oracle probes `info` for an entry containing $(\text{usk}_{\text{id}}, \text{cert}_{\text{id}})$ of user id , and returns $\chi \leftarrow \text{Claim}(\text{gpk}, (\text{usk}_{\text{id}}, \text{cert}_{\text{id}}), (M, \Sigma))$. It then sets $\text{CLAIM} \leftarrow \text{CLAIM} \parallel (\text{id}, (M, \Sigma), \chi)$.
- $\mathcal{O}_{\text{Open}}$: receives query on a message M and a signature Σ , the oracle verifies if (M, Σ) is valid. If not, it returns $\perp$. Otherwise, it returns the output of $\text{Open}(\text{gpk}, \text{osk}, (M, \Sigma))$.

For an oracle $\mathcal{O}$, we denote $\mathcal{O}^{\neg S}$ to indicate that $\mathcal{O}$ rejects queries on any input in the set S . We remark that from the above description, an adversary can run $\mathcal{O}_{\text{Sign}}$, $\mathcal{O}_{\text{Reveal}}$ and $\mathcal{O}_{\text{Claim}}$ by itself on any controlled users. For such reason, these oracles only accept queries on identifiers that belong to honest users.

For security notion, a traceable signature scheme should be traceable, non-frameable and anonymous.

Traceability. This is captured by *misidentification attacks*, in which an adversary attempts to create a non-trivial valid signature that is not opened to any of controlled users, or cannot be traced back to one of them. The adversary can control some group members and even some tracing agents.

Definition 15 (Traceability [30, 34]). A TS scheme $\mathcal{TS}$ is said to be traceable if for all polynomial $\lambda(\cdot)$ and all probabilistic, polynomial time (PPT) adversaries $\mathcal{A}$, its advantage $\text{Adv}_{\mathcal{TS}, \mathcal{A}}^{\text{trace}}(\lambda) := \Pr[\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{trace}}(\lambda) = 1]$ is negligible in the security parameter λ . The experiment $\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{trace}}$, formalizing misidentification attacks, is depicted as in Figure A.1.

```

1  pp  $\leftarrow$  Setup( $1^\lambda, N$ )
2  (gpk, gsk, osk)  $\leftarrow$  KeyGen( $1^\lambda$ )
3   $(M^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pJoin}}, \mathcal{O}_{\text{aJoin}}, \mathcal{O}_{\text{Corrupt}}, \mathcal{O}_{\text{Sign}}, \mathcal{O}_{\text{Reveal}}}(\text{gpk})$ 
4  if Verify(gpk,  $M^*, \Sigma^*$ ) = 0 then
5  |   return 0
6  if  $\exists \text{id} \in \text{Honest} : (\text{id}, M^*, \Sigma^*) \in \text{SIGN}$  then
7  |   return 0
8  if Open(gpk, osk,  $(M^*, \Sigma^*)$ )  $\notin$  Control or  $\forall \text{id} \in \text{Control} : \text{Trace}(\text{gpk}, (M^*, \Sigma^*), \text{Reveal}(\text{gpk}, \text{osk}, \text{id})) = 0$ 
   then
9  |   return 1
10 return 0

```

Figure A.1: Experiment $\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{trace}}(\lambda)$ defining traceability.

Non-frameability. This is captured by *framing attacks*, in which an adversary aims to 1) produce a valid signature that is opened or traced to an honest user; or 2) forge a claim of a signature generated by honest user. In this type of attacks, adversary can corrupt both of the $\mathcal{GM}$ and $\mathcal{OA}$.

Definition 16 (Non-frameability [30, 39]). A TS scheme $\mathcal{TS}$ is said to be non-frameable if for all polynomial $\lambda(\cdot)$ and all PPT adversaries $\mathcal{A}$, its advantage $\text{Adv}_{\mathcal{TS}, \mathcal{A}}^{\text{frame}}(\lambda) := \Pr[\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{frame}}(\lambda) = 1]$ is negligible in the security parameter λ . The experiment $\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{frame}}$, formalizing framing attacks, is depicted as in Figure A.2.

```

1   $\text{pp} \leftarrow \text{Setup}(1^\lambda, N)$ 
2   $(\text{gpk}, \text{gsk}, \text{osk}) \leftarrow \text{KeyGen}(1^\lambda)$ 
3   $(M^*, \Sigma^*, \chi^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pJoin}}, \mathcal{O}_{\text{Corrupt}}, \mathcal{O}_{\text{Sign}}}(\text{gpk}, \text{gsk}, \text{osk})$ 
4  if  $\text{Verify}(\text{gpk}, M^*, \Sigma^*) = 0$  then
5    | return 0
6  if
     $(\text{Open}(\text{gpk}, \text{osk}, (M^*, \Sigma^*)) \in \text{Honest}) \vee (\exists \text{id} \in \text{Honest} : \text{Trace}(\text{gpk}, (M^*, \Sigma^*), \text{Reveal}(\text{gpk}, \text{osk}, \text{id})) = 1)$ 
    and  $(\cdot, M^*, \Sigma^*) \notin \text{SIGN}$  then
7    | return 1
8  if  $\exists \text{id}' \in \text{Honest} : (\text{id}', M^*, \Sigma^*) \in \text{SIGN}$  and  $\text{ClaimVerify}(\text{gpk}, (M^*, \Sigma^*), \chi^*) = 1$  then
9    | return 1
10 return 0

```

Figure A.2: Experiment $\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{frame}}(\lambda)$ defining non-frameability.

In Figure A.2, line 8 captures the winning condition when the adversary successfully creates a claim of a signature created by an honest user id' , without knowing id' 's secret.

Anonymity. This is captured via a 2-phase game. During the first phase, an adversary is allowed to introduce honest and controlled users to the system, observe generated signatures and control some of the tracing agents. Then it chooses a challenge message M^* and two honest users id_0^* and id_1^* of which the tracing trapdoors are not revealed. The phase concludes when adversary receives a signature Σ^* which is signed by either id_0^* or id_1^* . The adversary continues its interaction with the oracles in second phase and attempts to guess which of the targeted users creates Σ^* .

Naturally, Definition 17 captures anonymity in the strongest sense (CCA-anonymity) by providing the adversary access to opening oracle. If this is not the case, we obtain a weaker notion of anonymity as in [30, 34].

In the experiment defining anonymity, unlike [35] we do not provide adversary with a tracing oracle. This is because tracing can be done implicitly if access to opening oracle is allowed. In addition, $\mathcal{O}_{\text{Reveal}}$ already models the threat of corrupted tracing agents (as discussed in [34]).

Definition 17 (Anonymity [30, 34, 35]). *A TS scheme $\mathcal{TS}$ is CCA-anonymous if for all polynomial $\lambda(\cdot)$ and all PPT adversaries $\mathcal{A}$, its advantage*

$$\text{Adv}_{\mathcal{TS}, \mathcal{A}}^{\text{anon}}(\lambda) := |\Pr[\text{Exp}_{\mathcal{TS}, \mathcal{A}}^{\text{anon}}(\lambda) = 1] - 1/2|$$

is negligible in the security parameter λ . The experiment $\text{Exp}_{\mathcal{T}, \mathcal{A}}^{\text{anon}}$ is depicted as in Figure A.3.

```

1  pp  $\leftarrow$  Setup( $1^\lambda, N$ )
2  (gpk, gsk, osk)  $\leftarrow$  Setup( $1^\lambda$ )
3  ( $st, M^*, \text{id}_0^*, \text{id}_1^*$ )  $\leftarrow \mathcal{A}^{\mathcal{O}_{\text{pJoin}}, \mathcal{O}_{\text{Corrupt}}, \mathcal{O}_{\text{Sign}}, \mathcal{O}_{\text{Reveal}}, \mathcal{O}_{\text{Claim}}, \mathcal{O}_{\text{Open}}}(\text{gpk}, \text{gsk})$ 
4  if  $\text{id}_0^* \notin \text{Honest}$  or  $\text{id}_0^* \in \text{REV}$  or  $\text{id}_1^* \notin \text{Honest}$  or  $\text{id}_1^* \in \text{REV}$  then
5  |   return 0
6   $b \xleftarrow{\$} \{0, 1\}$ 
7   $\Sigma^* \leftarrow \text{Sign}(\text{gpk}, (\text{sk}_{\text{id}_b^*}, \text{cert}_{\text{id}_b^*}), M^*)$ 
8   $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{pJoin}}, \mathcal{O}_{\text{Corrupt}}^{-\{\text{id}_0^*, \text{id}_1^*\}}, \mathcal{O}_{\text{Sign}}, \mathcal{O}_{\text{Reveal}}^{-\{\text{id}_0^*, \text{id}_1^*\}}, \mathcal{O}_{\text{Claim}}^{-\{(\cdot, M^*, \Sigma^*)\}}, \mathcal{O}_{\text{Open}}^{-\{M^*, \Sigma^*\}}}(\text{gpk}, \text{gsk}, st)$ 
9  if  $b = b'$  then
10 |   return 1
11 return 0

```

Figure A.3: Experiment $\text{Exp}_{\mathcal{T}, \mathcal{A}}^{\text{anon}}(\lambda)$ defining CCA-anonymity, the adversary is not allowed to make trivial query to reveal/claiming/opening oracles.

Appendix B. NIZK in QROM

We recall some results regarding NIZK proofs/arguments in QROM in [48].

Appendix B.1. Sigma Protocol

A Sigma protocol Σ for a relation $\mathcal{R}$ is a tuple $(D_{\text{com}}, D_{\text{ch}}, D_{\text{rsp}}, P_{\Sigma}^1, P_{\Sigma}^2, V_{\Sigma})$; in which $D_{\text{com}}, D_{\text{ch}}, D_{\text{rsp}}$ are domains of first, second and third message respectively; $(P_{\Sigma}^1, P_{\Sigma}^2)$ are two PPT algorithms that share a shame state and together form a prover $\mathcal{P}$; V_{Σ} is a deterministic, PT algorithm constituting a verifier $\mathcal{V}$ and outputting a single bit. On input $(x, w) \in \mathcal{R}$ where x is a statement and w is a corresponding witness, $P_{\Sigma}^1(x, w) \rightarrow \text{com}$, where the first message com is called *commitment* and is sent to $\mathcal{V}$. The second message is called *challenge*, denoted by ch and is uniformly chosen from D_{ch} by $\mathcal{V}$ and is sent to the $\mathcal{P}$. The final message is called *response*, sent by $\mathcal{P}$ to $\mathcal{V}$ as $\text{rsp} \leftarrow P_{\Sigma}^2(x, w, \text{com}, \text{ch})$. Finally, $\mathcal{V}$ runs V_{Σ} on input $(x, \text{com}, \text{ch}, \text{rsp})$ and accepts iff V_{Σ} outputs 1. The tuple $(\text{com}, \text{ch}, \text{rsp})$ is called a transcript of Σ .

A sigma protocol should satisfy the following properties:

1. **Completeness:** If $\mathcal{P}$ and $\mathcal{V}$ follow the protocol with statement x as the common input and witness w as the private input to $\mathcal{P}$, such that $(x, w) \in \mathcal{R}$; then $\mathcal{V}$ accepts with overwhelming probability.

2. **(Computational) k -special soundness:** Given a positive integer $k \geq 2$ and let $(\text{com}, \text{ch}_1, \text{rsp}_1), \dots, (\text{com}, \text{ch}_k, \text{rsp}_k)$ be k valid transcripts of Σ for a statement x , produced by some quantum PT algorithm $\mathcal{A}$, where $\text{ch}_1, \dots, \text{ch}_k$ are pairwise distinct. Then there exists a PPT algorithm E_Σ called *knowledge extractor*, taking k valid transcripts as input and return a witness w such that $(x, w) \in \mathcal{R}$ with overwhelming probability.
3. **Honest verifier zero-knowledge (HVZK):** There exists a PPT algorithm S_Σ , called *simulator*, such that for all PPT quantum algorithm $\mathcal{A}$ and all $(x, w) \in \mathcal{R}$, the following quantity is negligible

$$|\Pr[b = 1 : (\text{com}, \text{ch}, \text{rsp}) \leftarrow (\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x)), b \leftarrow \mathcal{A}(\text{com}, \text{ch}, \text{rsp})] - \Pr[b = 1 : (\text{com}, \text{ch}, \text{rsp}) \leftarrow S_\Sigma(x), b \leftarrow \mathcal{A}(\text{com}, \text{ch}, \text{rsp})]|.$$

Here $(\text{com}, \text{ch}, \text{rsp}) \leftarrow (\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x))$ denotes the transcript obtained by executing protocol honestly with prover $\mathcal{P}$ and verifier $\mathcal{V}$.

Appendix B.2. NIZK protocol in QROM

A non-interactive proof protocol $\Psi = (\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ can be described as $(\pi \leftarrow \mathcal{P}^\mathcal{H}(x, w), b \leftarrow \mathcal{V}^\mathcal{H}(x, \pi))$. The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ are PPT algorithms, $\mathcal{H}$ is a random oracle, x is the statement, w is the witness such that $(x, w) \in \mathcal{R}$ and $b \in \{0, 1\}$. Denote by R the uniform distribution of random oracles.

For a non-interactive protocol Ψ , the following should be satisfied:

1. **Completeness:** If $\mathcal{P}$ and $\mathcal{V}$ follow the protocol with statement x as the common input and witness w as the private input to $\mathcal{P}$, such that $(x, w) \in \mathcal{R}$; then $\mathcal{V}$ outputs with overwhelming probability.
2. **Zero-knowledge:** There is a pair of algorithms (S_P, S_O) (called simulator), such that for any quantum polynomial-time algorithm $\mathcal{A}$, we have that

$$|\Pr[b = 1 : \mathcal{H} \leftarrow R, b \leftarrow \mathcal{A}^{\mathcal{H}, \mathcal{P}}()] - \Pr[b = 1 : \mathcal{H} \leftarrow S_O, b \leftarrow \mathcal{A}^{\mathcal{H}, S_P}()]|$$

is negligible, where $\mathcal{A}^{\mathcal{H}, \mathcal{P}}$ denotes the quantum PT algorithm $\mathcal{A}$ with oracle access to $\mathcal{H}$ and $\mathcal{P}$ (similar for $\mathcal{A}^{\mathcal{H}, S_P}$).

Another property is *online-extractability*, which is a special case of *special soundness*. Roughly speaking, online-extractability captures the ability to extract a witness without rewinding a prover. In QROM, an extractor against this property is allowed to get the circuit description of random oracles, which are generated by the random oracle simulator S_O of zero-knowledge property.

Definition 18. Ψ is *online-extractable* with respect to a simulator S_O if there exist a PPT extractor $\mathcal{E}$ such that for any quantum, PT algorithm $\mathcal{A}$, the following probability

$$\Pr[b = 1 \wedge (x, w) \notin \mathcal{R} : \mathcal{H} \leftarrow S_O, (x, \pi) \leftarrow \mathcal{A}^\mathcal{H}, b \leftarrow \mathcal{P}^\mathcal{H}(x, \pi), w \leftarrow \mathcal{E}(\mathcal{H}, x, \pi)]$$

```

1 for  $i = 1, \dots, t$  do
2    $\text{com}_i \leftarrow P_\Sigma^1(x, w)$ 
3   for  $j = 1, \dots, m$  do
4      $\text{ch}_{i,j} \xleftarrow{\$} D_{\text{ch}} \setminus \{\text{ch}_{i,1}, \dots, \text{ch}_{i,j-1}\}$ 
5      $\text{rsp}_{i,j} \leftarrow P_\Sigma^2(x, w, \text{com}_i, \text{ch}_{i,j})$ 
6 for  $i = 1, \dots, t$  do
7   for  $j = 1, \dots, m$  do
8      $h_{i,j} = \mathcal{H}_2(\text{rsp}_{i,j})$ 
9 Parse  $J_1 | \dots | J_t := \mathcal{H}_1(x, (\text{com}_i)_{[t]}, (\text{ch}_{i,j})_{[t] \times [m]}, (h_{i,j})_{[t] \times [m]}),$  where  $J_i \in [m]$ 
10 for  $i = 1, \dots, t$  do
11    $\text{rsp}_i := \text{rsp}_{i, J_i}$ 
12  $\pi := ((\text{com}_i)_{[t]}, (\text{ch}_{i,j})_{[t] \times [m]}, (h_{i,j})_{[t] \times [m]}, (\text{rsp}_i)_{[t]})$ 
13 return  $\pi$ 

```

Figure B.4: A prover $\mathcal{P}$, generates an NIZK proof π for $(x, w) \in \mathcal{R}$.

is negligible.

Simulation-sound online-extractability is a special case of *online-extractability*, which essentially states a witness extractor succeeds with overwhelming probability, even against an adversary that adaptively gets many simulated proofs.

Definition 19. Ψ is *simulation-sound online-extractable*, if there exist a PPT extractor $\mathcal{E}$ such that for any quantum, PT algorithm $\mathcal{A}$, the following probability

$$\Pr[b = 1 \wedge (x, \pi) \notin \mathbb{X} \wedge (x, w) \notin \mathcal{R} : \\ \mathcal{H} \leftarrow S_O, (x, \pi) \leftarrow \mathcal{A}^{\mathcal{H}, S_P}, b \leftarrow \mathcal{P}^{\mathcal{H}}(x, \pi), w \leftarrow \mathcal{E}(\mathcal{H}, x, \pi)]$$

is negligible. Here $\mathbb{X}$ denotes the set of proofs produced by S_P .

Appendix B.3. NIZK from Generalized Unruh Transformation

Let $\Sigma = (D_{\text{com}}, D_{\text{ch}}, D_{\text{rsp}}, P_\Sigma^1, P_\Sigma^2, V_\Sigma)$ be a sigma protocol for a relation $\mathcal{R}$ with completeness, k -special soundness and HVZK properties. The generalized Unruh transformation [48] transform Σ into a non-interactive protocol Ψ between prover $\mathcal{P}$ and verifier $\mathcal{V}$ by utilizing two hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$ modeled as random oracles, where $\mathcal{H}_1 : \{0, 1\}^* \rightarrow \{1, 2, \dots, m\}^t$ and $\mathcal{H}_2 : D_{\text{rsp}} \rightarrow D_{\text{rsp}}$. Here t is the number of times prover $\mathcal{P}$ needs to execute P_Σ^1 and m is a positive integer such that $m > k$. The algorithms of prover $\mathcal{P}$ and verifier $\mathcal{V}$ is depicted as in Figure 4 and Figure 5 respectively. Here, we let $[n]$ be the set $\{1, 2, \dots, n\}$.

```

1 Parse  $J_1 | \dots | J_t := \mathcal{H}_1(x, (\text{com}_i)_{[t]}, (\text{ch}_{i,j})_{[t] \times [m]}, (h_{i,j})_{[t] \times [m]})$ , where  $J_i \in [m]$ 
2 for  $i = 1, \dots, t$  do
3   | Verify that  $\text{ch}_{i,1}, \dots, \text{ch}_{i,t}$  pairwise distinct
4 for  $i = 1, \dots, t$  do
5   | Verify that  $V_\Sigma(x, \text{com}_i, \text{ch}_{i,j}, \text{rsp}_i) = 1$ 
6 for  $i = 1, \dots, t$  do
7   | Verify that  $h_{i,J_i} = \mathcal{H}_2(\text{rsp}_{i,j})$ 
8 if all checks succeed then
9   | return 1
10 else
11   | return 0

```

Figure B.5: Verifier $\mathcal{V}$ checks the validity of a proof π on statement x .

Proposition 3 ([48]). *Let Σ be a Sigma protocol that is complete, k -special soundness and HVZK. Then the non-interactive protocol Ψ derived from Σ as described in **Fig. 4** and **Fig. 5** is complete, zero-knowledge and simulation-sound online-extractable.*

To turn Ψ into a signature of knowledge on a message M , we simply include M as an input to $\mathcal{H}_1$. That is, in the prover's algorithm in **Fig. 4**, $\mathcal{P}$ computes

$$\mathcal{H}_1(M, x, (\text{com}_i)_{[t]}, (\text{ch}_{i,j})_{[t] \times [m]}, (h_{i,j})_{[t] \times [m]}).$$

In the case Σ has a common reference string crs , the NIZK prover computes

$$\mathcal{H}_1(\text{crs}, M, x, (\text{com}_i)_{[t]}, (\text{ch}_{i,j})_{[t] \times [m]}, (h_{i,j})_{[t] \times [m]}).$$

Appendix C. Lattice-Based ZK Argument for Quadratic Relation

We recall the ZK argument system from [46]. Let n, m, ℓ be positive integer. Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{x} \in \mathbb{Z}_q^n$ and $\mathbf{y} \in \mathbb{Z}_q^m$. Let $\mathcal{S}$ be a set of tuples (h, i, j) of integers in $[1, n]$ such that $|\mathcal{S}| = \ell$. The ZK argument system from [46] is a proof-of-knowledge protocol for the relation

$$\mathcal{R} = \{((\mathbf{A}, \mathbf{y}, \mathcal{S}), \mathbf{x}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q \wedge \forall (h, i, j) \in \mathcal{S} : \mathbf{x}[h] = \mathbf{x}[i] \cdot \mathbf{x}[j] \bmod q\}, \quad (\text{C.1})$$

The basic protocol Let λ be a security parameter and AuxCom be an arbitrary secure string commitment scheme with randomness space $\{0, 1\}^\kappa$. Let $l_1, l_2 = \Theta(\lambda)$ be positive integers; $\mathbf{B}_1$ and $\mathbf{B}_2$ are two random matrices in the form

$$\mathbf{B}_1 = \left(\begin{array}{c|cc} \mathbf{I}_{l_1} & & \mathbf{B}_{1,1} \\ \hline \mathbf{0}^{n \times l_1} & \mathbf{I}_n & \mathbf{B}_{1,2} \end{array} \right), \quad \mathbf{B}_2 = \left(\begin{array}{c|cc} \mathbf{I}_{l_1} & & \mathbf{B}_{2,1} \\ \hline \mathbf{0}^{\ell \times l_1} & \mathbf{I}_\ell & \mathbf{B}_{2,2} \end{array} \right),$$

where $\mathbf{B}_{1,1}, \mathbf{B}_{1,2}, \mathbf{B}_{2,1}, \mathbf{B}_{2,2}$ are uniformly random matrices sampled from $\mathbb{Z}_q^{l_1 \times (l_2+n)}, \mathbb{Z}_q^{n \times l_2}, \mathbb{Z}_q^{l_1 \times (l_2+\ell)}$ and $\mathbb{Z}_q^{\ell \times l_2}$ respectively. Looking forward, $\mathbf{B}_1$ and $\mathbf{B}_2$ are public parameters of the BDLOP commitment scheme [62].

In addition, the system choose a small integer $p = \text{poly}(\lambda)$ defining the soundness error. The randomnesses for BDLOP commitment scheme are sampled from two discrete Gaussian distributions, with width $\sigma_1 \geq \sqrt{2l_2/\pi}$ and $\sigma_2 = 2 \cdot p \cdot l \cdot \log l \cdot \sigma_1$, where $l = 2l_1 + 2l_2 + n + \ell$. The system modulus q is a power of a prime q_0 , i.e. $q = q_0^e$ and is chosen such that $q \geq 16p \cdot \max\{l_1 + l_2 + n, l_1 + l_2 + \ell\} \cdot (\sigma_2 + p \cdot \sigma_1) \cdot \tilde{\mathcal{O}}(\sqrt{l_1})$ and q/σ_1 is a polynomial. The common reference string is defined as

$$\text{crs} = (\text{AuxCom}, \mathbf{B}_1, \mathbf{B}_2, \sigma_1, \sigma_2, p).$$

In the following protocol, let $\mathbf{p}(\mathbf{v}, \mathbf{z}) = \min \left(1, \frac{D_{\sigma_2}^l(\mathbf{z})}{M \cdot D_{\sigma_2, \mathbf{v}}^l(\mathbf{z})} \right)$ for any vectors $\mathbf{v}, \mathbf{z} \in \mathbb{Z}^l$, where $M = e^{\mathcal{O}(1/\log^2 l)}$.

Σ -protocol for relation $\mathcal{R}$

1. **Commitment:**

- $\mathcal{P}$ samples uniformly random $\mathbf{r} \in \mathbb{Z}_q^n$, computes $\mathbf{t} \leftarrow \mathbf{A} \cdot \mathbf{r} \bmod q$;
- $\mathcal{P}$ samples randomnesses $\mathbf{s}_1 \leftarrow D_{\sigma_1}^{l_2+n+l_1}$, $\mathbf{s}_2 \leftarrow D_{\sigma_2}^{l_2+n+l_1}$, $\mathbf{s}_3 \leftarrow D_{\sigma_1}^{l_2+\ell+l_1}$, $\mathbf{s}_4 \leftarrow D_{\sigma_2}^{l_2+\ell+l_1}$;
- $\mathbf{c}_1 \leftarrow \mathbf{B}_1 \cdot \mathbf{s}_1 + (\mathbf{0}^T | \mathbf{x}^T)^T \bmod q$, $\mathbf{c}_2 \leftarrow \mathbf{B}_1 \cdot \mathbf{s}_2 + (\mathbf{0}^T | \mathbf{r}^T)^T \bmod q$;
- Let $\mathbf{a}, \mathbf{b}$ be two vectors in $\mathbb{Z}_q^\ell$;
- For $k \in [1, \ell]$, let (h, i, j) be the k -th element of $\mathcal{S}$ and compute

$$\mathbf{a}[k] = \mathbf{r}[h] - \mathbf{r}[i] \cdot \mathbf{r}[j] - \mathbf{r}[j] \cdot \mathbf{x}[i], \quad \mathbf{b}[k] = \mathbf{r}[i] \cdot \mathbf{r}[j];$$

- $\mathbf{c}_3 \leftarrow \mathbf{B}_2 \cdot \mathbf{s}_3 + (\mathbf{0}^T | \mathbf{a}^T)^T$, $\mathbf{c}_4 \leftarrow \mathbf{B}_2 \cdot \mathbf{s}_4 + (\mathbf{0}^T | \mathbf{b}^T)^T$;
- Samples randomness $\rho \xleftarrow{\$} \{0, 1\}^\kappa$, commits

$$\text{com}_{\text{aux}} \leftarrow \text{AuxCom}(\mathbf{t} \mid \mathbf{c}_1 \mid \mathbf{c}_2 \mid \mathbf{c}_3 \mid \mathbf{c}_4; \rho)$$

and sends com_{aux} to $\mathcal{V}$.

2. **Challenge:** $\mathcal{V}$ sends a uniformly random challenge $\text{ch} \in [-p, p]$ to $\mathcal{P}$.

3. **Response:** $\mathcal{P}$ computes

$$\mathbf{z}_0 \leftarrow \text{ch} \cdot \mathbf{x} + \mathbf{r}, \quad \mathbf{z}_1 \leftarrow \text{ch} \cdot \mathbf{s}_1 + \mathbf{s}_2, \quad \mathbf{z}_2 \leftarrow \text{ch} \cdot \mathbf{s}_3 - \mathbf{s}_4;$$

aborts with probability $1 - \mathbf{p}((\text{ch} \cdot \mathbf{s}_1^T | \text{ch} \cdot \mathbf{s}_2^T)^T, (\mathbf{z}_1^T | \mathbf{z}_2^T)^T)$, and sends the response as $\text{rsp} = (\mathbf{c}_1, \mathbf{c}_3, \rho, \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2)$.

Verification: Receiving rsp , $\mathcal{V}$ parses $\text{rsp} = (\mathbf{c}_1, \mathbf{c}_3, \rho, \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2)$ and proceeds as follows:

- Let $\mathbf{d} \in \mathbb{Z}^\ell$. For $k \in [1, \ell]$, let (h, i, j) be the k -th element of $\mathcal{S}$ and set

$$\mathbf{d}[k] \leftarrow \text{ch} \cdot \mathbf{z}_0[h] - \mathbf{z}_0[i] \cdot \mathbf{z}_0[j];$$

- Set

$$\mathbf{t} \leftarrow \mathbf{A} \cdot \mathbf{z}_0 - \text{ch} \cdot \mathbf{y};$$

$$\mathbf{c}_2 \leftarrow \mathbf{B}_1 \cdot \mathbf{z}_1 + (\mathbf{0}^T | \mathbf{z}_0^T)^T - \text{ch} \cdot \mathbf{c}_1;$$

$$\mathbf{c}_4 \leftarrow \text{ch} \cdot \mathbf{c}_3 - \mathbf{B}_2 \cdot \mathbf{z}_2 + (\mathbf{0}^T | \mathbf{d}_0^T)^T;$$

- Verify that:

$$- \text{com}_{\text{aux}} = \text{AuxCom}(\mathbf{t} | \mathbf{c}_1 | \mathbf{c}_2 | \mathbf{c}_3 | \mathbf{c}_4; \rho);$$

$$- \|\mathbf{z}_1\| \leq 2\sqrt{l_1 + l_2 + n} \cdot (\sigma_2 + p \cdot \sigma_1);$$

$$- \|\mathbf{z}_2\| \leq 2\sqrt{l_1 + l_2 + \ell} \cdot (\sigma_2 + p \cdot \sigma_1).$$

$\mathcal{V}$ outputs 1 if and only if all the conditions hold.

In the above protocol, response and verification algorithms slightly differ from the original one. In fact, these are the tweaks proposed in [46] for the non-interactive version. They help reduce the communication cost and do not affect the overall security.

Proposition 4 (Adapted from [46]). *Suppose that $p < q_0/2$, then the above protocol is a ZK argument protocol for relation $\mathcal{R}$ with completeness error $1 - 1/M$, soundness error at most $2/(2p + 1)$ and is honest-verifier zero knowledge assuming the hardness of $\text{SIS}_{l_1, l_1 + l_2 + n, q, \beta_1}^{(2)}$, $\text{SIS}_{l_1, l_1 + l_2 + \ell, q, \beta_2}^{(2)}$ and $\text{LWE}_{l_2, q, \mathcal{D}_{\sigma_1}}$, where $\beta_1 = 16p \cdot \sqrt{l_1 + l_2 + n} \cdot (\sigma_2 + p \cdot \sigma_1)$ and $\beta_2 = 16p \cdot \sqrt{l_1 + l_2 + \ell} \cdot (\sigma_2 + p \cdot \sigma_1)$. In particular, the prover's response has bit size at most $\kappa + (6l_1 + 2l_2 + 3n + 2\ell) \log q$ and the protocol is 3-soundness: given 3 valid pairs of challenge-response with the same commitment, there is a PPT extractor for a witness of $(\mathbf{A}, \mathbf{y}, \mathcal{S})$.*

We remark that the protocol has non-negligible completeness error due to aborting, therefore when apply the algorithm in **Fig. 4** one keeps repeating the protocol until all responses are generated successfully. The number repetitions can be reduced by employing rejection sampling technique. Additionally, AuxCom should be a binding and hiding string commitment scheme. In lattice settings, it can be instantiated by the KTX commitment scheme [65], which is computationally binding (under SIS assumption) and statistically hiding.

Appendix D. Detailed NIZK Proofs for Lattice-Based Traceable Signature Scheme

We explain how to prove the relations $\mathcal{R}_{\text{Sign}}$ in Definition 8 and $\mathcal{R}_{\text{Claim}}$ in Definition 9. First, we consider a generic relation that captures all the relations defining $\mathcal{R}_{\text{Sign}}$ and $\mathcal{R}_{\text{Claim}}$. Then we give the detail transformation of each of those relations to a case of (C.1).

Appendix D.1. Proving a Generic Relation

Recall that the TS scheme in Section 3.2 works with two moduli q and q' , where $q > q'$. As the relations defining our TS scheme involves equations modulo q' , we need to appropriately transform them into sets of equations modulo q .

Following the observation in [66], statements (i), (iii) and (iv) are special cases of the following relation

$$\mathcal{R}_{m,n,q',\beta} = \{(\mathbf{A}, \mathbf{y}, \mathbf{x}) \in \mathbb{Z}_{q'}^{m \times n} \times \mathbb{Z}_{q'}^m \times \mathbb{Z}_{q'}^n : \mathbf{y} = \mathbf{A} \cdot \mathbf{x} \bmod q'; \|\mathbf{x}\|_\infty \leq \beta\} \quad (\text{D.1})$$

where $1 \leq \beta \leq (q' - 1)/2$. We show that relation (D.1) can be transformed into a case of relation (C.1). Rewrite the equation modulo q' as $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} + q' \cdot \mathbf{v}$ for some $\mathbf{v} \in \mathbb{Z}^m$. Observe that

$$q' \cdot \|\mathbf{v}\|_\infty \leq (\|\mathbf{y}\|_\infty + \|\mathbf{A} \cdot \mathbf{x}\|_\infty) \leq q' \cdot \frac{n\beta + 1}{2},$$

since the entries of $\mathbf{A}$ have absolute values at most $(q' - 1)/2$. Thus, for sufficiently large q , namely $q > q'(n\beta + 1)$, if we let $\mathbf{A}' = [\mathbf{A}|q'\mathbf{I}_m] \in \mathbb{Z}_q^{m \times (n+m)}$, $\mathbf{x}' = [\mathbf{x}|\mathbf{v}] \in \mathbb{Z}_q^{n+m}$ and $\beta' = (n\beta + 1)/2$, then $\mathcal{R}_{m,n,q',\beta}$ is equivalent to the following relation

$$\begin{aligned} \mathcal{R}_{n,m,q,\beta,\beta'} &= \{((\mathbf{A}', \mathbf{y}), \mathbf{x}') \in \mathbb{Z}_q^{m \times (n+m)} \times \mathbb{Z}_q^m \times \mathbb{Z}_q^{n+m} : \mathbf{y} = \mathbf{A}' \cdot \mathbf{x}' \bmod q; \\ &\quad \mathbf{x}' = [\mathbf{x}|\mathbf{v}] \in \mathbb{Z}_q^{n+m}; \|\mathbf{x}\|_\infty \leq \beta; \|\mathbf{v}\|_\infty \leq \beta'\}, \end{aligned} \quad (\text{D.2})$$

since $\|\mathbf{A}'\mathbf{x}'\|_\infty < q/2$. Now we transform the relation $\mathcal{R}_{n,m,q,\beta,\beta'}$ to a case of relation $\mathcal{R}$ in (C.1), by applying the binary decomposition technique from [61]. Formally, let $\mathbf{b} = (\beta \dots \beta)^T \in \mathbb{Z}_q^n$, $\mathbf{b}' = (\beta' \dots \beta')^T \in \mathbb{Z}_q^m$ and let $\mathbf{x}_1 = \mathbf{x} + \mathbf{b}$, $\mathbf{x}_2 = \mathbf{v} + \mathbf{b}'$. Let $k = \lceil \log(2\beta) \rceil + 1$ and

$$\mathbf{g}_1 = \left(\left\lfloor \frac{2\beta + 1}{2} \right\rfloor \quad \left\lfloor \frac{2\beta + 2}{4} \right\rfloor \quad \dots \quad \left\lfloor \frac{2\beta + 2^{k-1}}{2^k} \right\rfloor \right) \in \mathbb{Z}^k$$

be a row vector. In [61], it is proven that an integer $a \in [0, 2\beta]$ if and only if there exists a binary vector $\mathbf{a} \in \{0, 1\}^k$ such that $\mathbf{g}_1 \cdot \mathbf{a} = a$. Using this fact, if we let $\mathbf{G}_1 = \mathbf{I}_n \otimes \mathbf{g}_1$, then there exists a binary vector $\overline{\mathbf{x}}_1 \in \{0, 1\}^{kn}$ such that $\mathbf{G}_1 \cdot \overline{\mathbf{x}}_1 = \mathbf{x}_1$. Similarly, let

$$\begin{aligned} k' &= \lceil \log(2\beta') \rceil + 1; \\ \mathbf{g}_2 &= \left(\left\lfloor \frac{2\beta' + 1}{2} \right\rfloor \quad \left\lfloor \frac{2\beta' + 2}{4} \right\rfloor \quad \dots \quad \left\lfloor \frac{2\beta' + 2^{k'-1}}{2^{k'}} \right\rfloor \right) \in \mathbb{Z}^{k'}; \\ \mathbf{G}_2 &= \mathbf{I}_m \otimes \mathbf{g}_2 \in \mathbb{Z}^{m \times k'm}; \end{aligned}$$

then there exists a binary vector $\bar{\mathbf{x}}_2 \in \{0, 1\}^{k'm}$ such that $\mathbf{G}_2 \cdot \bar{\mathbf{x}}_2 = \mathbf{x}_2$. Now, form the matrix $\bar{\mathbf{A}}' = [\mathbf{A} \cdot \mathbf{G}_1 | q' \mathbf{I}_m \cdot \mathbf{G}_2] \in \mathbb{Z}_q^{m \times (kn+k'm)}$ and form the column vector $\bar{\mathbf{x}}' = [\bar{\mathbf{x}}_1^T | \bar{\mathbf{x}}_2^T]^T \in \{0, 1\}^{kn+k'm}$. The relation $\mathcal{R}_{n,m,q,\beta,\beta'}$ is transformed into

$$\{((\bar{\mathbf{A}}', \mathbf{y}), \bar{\mathbf{x}}') \in \mathbb{Z}_q^{m \times (kn+k'm)} \times \mathbb{Z}_q^m \times \{0, 1\}^{kn+k'm} : \mathbf{y} = \bar{\mathbf{A}}' \cdot \bar{\mathbf{x}}' \bmod q\}. \quad (\text{D.3})$$

This is a linear relation modulo q , with a binary constraint over the witness $\bar{\mathbf{x}}'$. Note that when q is prime and $x \in \mathbb{Z}_q$, then we have $x \in \{0, 1\}$ iff $x = x^2 \bmod q$. Hence, if we let $\mathcal{S} = \{(i, i, i) : i \in \{1, 2, \dots, kn+k'm\}\}$, then the above relation can be rewritten into the following

$$\{((\bar{\mathbf{A}}', \mathbf{y}, \mathcal{S}), \bar{\mathbf{x}}') : \mathbf{y} = \bar{\mathbf{A}}' \cdot \bar{\mathbf{x}}' \bmod q \wedge \forall (i, i, i) \in \mathcal{S} : \bar{\mathbf{x}}'[i] = \bar{\mathbf{x}}'[i] \cdot \bar{\mathbf{x}}'[i]\}.$$

which is a case of the quadratic relation in (C.1). It follows that one can prove relation (D.1) by Yang *et al* argument [46].

Appendix D.2. Proving the Defining Statements of Lattice-Based Traceable Signature Scheme

In this section we describe the transformations of the statements (i), (iii), (iv) and (v) into a case of relation (D.1).

Proof of Image-Preimage of SIS One-way Function. Let $\mathbf{F} \in \mathbb{Z}_{q'}^{n \times m_{\mathbf{F}}}$ define an SIS-based one-way function over domain $\{0, 1\}^{m_{\mathbf{F}}}$ and range $\mathbb{Z}_{q'}^n$. We want to prove in zero-knowledge the following relation

$$\mathcal{R}_{\text{SIS}} = \left\{ (\mathbf{F}, (\mathbf{x}, \mathbf{z})) \in \mathbb{Z}_{q'}^{n \times m_{\mathbf{F}}} \times \mathbb{Z}_q^n \times \{0, 1\}^{m_{\mathbf{F}}} : \mathbf{x} = \mathbf{F} \cdot \mathbf{z} \bmod q' \right\}.$$

Let $\mathbf{w} = \text{bin}(\mathbf{x}) \in \{0, 1\}^k$, where $k = \lceil \log q' \rceil$ and $\mathbf{G}_n \in \mathbb{Z}_q^{n \times k}$ be a gadget matrix. It follows that $\mathbf{G}_n \cdot \mathbf{w} = \mathbf{x}$ and that $\mathbf{G}_n \cdot \mathbf{w} = \mathbf{F} \cdot \mathbf{s} - q' \cdot \mathbf{a}$ for $\mathbf{a} \in \mathbb{Z}^{m_{\mathbf{F}}}$. In addition

$$q' \|\mathbf{a}\|_{\infty} \leq (\|\mathbf{t}\|_{\infty} + \|\mathbf{F} \cdot \mathbf{s}\|_{\infty}) \leq q'(m_{\mathbf{F}} + 1)/2.$$

By choosing $q > q'(m_{\mathbf{F}} + 1)$, and letting

$$\begin{aligned} \bar{\mathbf{A}} &= \begin{pmatrix} \mathbf{G}_n & -\mathbf{F} & q' \mathbf{I}_{m_{\mathbf{F}}} \end{pmatrix} \in \mathbb{Z}^{n \times (2m_{\mathbf{F}}+k)}; \\ \bar{\mathbf{x}} &= (\mathbf{w}, \mathbf{z}, \mathbf{a}) \in \mathbb{Z}^{2m_{\mathbf{F}}+k}; \end{aligned}$$

then we have $\bar{\mathbf{A}} \cdot \bar{\mathbf{x}} = \mathbf{0} \bmod q$, where the components $\mathbf{w}, \mathbf{z}$ in $\bar{\mathbf{x}}$ are binary vectors and $\mathbf{a}$ has infinity norm at most $(m_{\mathbf{F}} + 1)/2$. Note that, $(\bar{\mathbf{A}}, \bar{\mathbf{x}})$ is an instance of relation (D.2). Hence we can apply the binary decomposition technique to $\mathbf{a}$ to transform $\mathcal{R}_{\text{SIS}}$ to a case of relation (C.1). Overall, the final quadratic relation has witness length and number of quadratic constraints approximately $m_{\mathbf{F}} + n \log q' + m_{\mathbf{F}} \log(m_{\mathbf{F}} + 1) \approx 2n \log q' + 3n \log(n \log q')$.

Proof of LWE Sample-Secret. Let $\mathbf{y} \in \mathbb{Z}_{q'}^{m_B}$, $\mathbf{x} \in \mathbb{Z}_{q'}^n$, $\mathbf{e} \in \mathbb{Z}^{m_B}$ and $\mathbf{B} \in \mathbb{Z}_{q'}^{m_B \times n}$, we need to prove in ZK the following relation

$$\mathcal{R}_{\text{LWE}}^{(1)} = \{(\mathbf{M}, (\mathbf{y}, \mathbf{x}, \mathbf{e})) : \mathbf{y} = \mathbf{B}^\top \cdot \mathbf{x} + \mathbf{e} \bmod q' \wedge \|\mathbf{e}\|_\infty \leq B_{\text{LWE}}\}.$$

Let $\mathbf{G}_{m_B}$ be a gadget matrix and $\text{bin}(\mathbf{y}) \in \{0, 1\}^{m_B \lceil \log q' \rceil}$, we have that $\mathbf{G}_{m_B} \cdot \text{bin}(\mathbf{y}) = \mathbf{y} \bmod q'$. Write $\mathbf{y} - \mathbf{B}^\top \cdot \mathbf{x} - \mathbf{e} = q' \cdot \mathbf{a}$ for $\mathbf{a} \in \mathbb{Z}^{m_B}$ then

$$q' \cdot \|\mathbf{a}\|_\infty \leq \|\mathbf{y} - \mathbf{B}^\top \cdot \mathbf{x} - \mathbf{e}\|_\infty \leq (q'/2) + (q'/2)^2 + B_{\text{LWE}}.$$

By choosing $q > q' + q'^2 + 2B_{\text{LWE}}$ and letting

$$\begin{aligned} \bar{\mathbf{A}} &= \begin{pmatrix} \mathbf{I}_{m_B} & -\mathbf{B}^\top & \mathbf{I}_{m_M} & \mathbf{0}^{m_B \times m_B \lceil \log q' \rceil} & -q' \mathbf{I}_{m_M} \\ -\mathbf{I}_{m_B} & \mathbf{0}^{m_B \times n} & \mathbf{0}^{m_B \times m_B} & \mathbf{G}_{m_B} & \mathbf{0}^{m_B \times m_B} \end{pmatrix} \in \mathbb{Z}_q^{2m_B \times (3m_B + n + m_B \lceil \log q' \rceil)}, \\ \bar{\mathbf{x}} &= (\mathbf{y}, \mathbf{x}, \mathbf{e}, \text{bin}(\mathbf{y}), \mathbf{a}) \in \mathbb{Z}^{m_B} \times \mathbb{Z}^n \times \mathbb{Z}^{m_B} \times \{0, 1\}^{m_B \lceil \log q' \rceil} \times \mathbb{Z}^{m_B}, \end{aligned}$$

we have that $\bar{\mathbf{A}} \cdot \bar{\mathbf{x}} = \mathbf{0} \bmod q$. By using the technique that transforms relation (D.1) to a case of relation (C.1), we can transform $\mathcal{R}_{\text{LWE}}^{(1)}$ to a case of relation (C.1) as well. The final quadratic relation has witness length and number of quadratic constraints approximately $3m_B \log(q') + n \log(q') + m_B \log(2B_{\text{LWE}})$.

We remark here that in addition to proving $\mathcal{R}_{\text{LWE}}^{(1)}$, we also prove the well-formedness of the binary message $\text{bin}(\mathbf{y})$ certified by the group manager (i.e. $\text{bin}(\mathbf{y})$ is a correct binary decomposition of $\mathbf{y}$), thus provide a link between statements (ii) and (iii).

Proof of Plaintext Knowledge in GPV-IBE. Let $\mathbf{c} \in \mathbb{Z}_{q'}^{m_B+1}$, $\mathbf{B} \in \mathbb{Z}_{q'}^{n \times m_B}$, $\mathbf{v} \in \mathbb{Z}_{q'}^n$, $\mathbf{r} \in \mathbb{Z}_{q'}^n$, $\mathbf{e}_c \in [-B_{\text{GPV}}, B_{\text{GPV}}]^{m_B+1}$ and $\text{id} \in \{1, \dots, N = 2^\ell - 1\}$. We need to prove in ZK the following relation

$$\mathcal{R}_{\text{Enc}} = \left\{ ((\mathbf{c}, \mathbf{B}, \mathbf{v}), (\mathbf{r}, \mathbf{e}_c, \text{id})) : \mathbf{c} = \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} \cdot \mathbf{r} + \mathbf{e}_c + \begin{pmatrix} \mathbf{0}^{m_B} \\ \lceil q'/(2(N+1)) \rceil \cdot \text{id} \end{pmatrix} \bmod q' \right\}.$$

Let

$$\mathbf{C}' = \begin{pmatrix} \begin{pmatrix} \mathbf{B}^\top \\ \mathbf{v}^\top \end{pmatrix} & \mathbf{I}_{m_B+1} & \begin{pmatrix} \mathbf{0}^{m_B} \\ \lceil q'/(2(N+1)) \rceil \end{pmatrix} \end{pmatrix} \in \mathbb{Z}_{q'}^{(m_B+1) \times (n+m_B+1)}$$

it follows that $\mathbf{c} = \mathbf{C}' \cdot (\mathbf{r}, \mathbf{e}_c, \text{id}) \bmod q'$. Write $\mathbf{c} = \mathbf{C}' \cdot (\mathbf{r}, \mathbf{e}_c, \text{id}) - q' \cdot \mathbf{a}$ for $\mathbf{a} \in \mathbb{Z}^{m_B+1}$, then

$$q' \cdot \|\mathbf{a}\|_\infty \leq \|\mathbf{c}\|_\infty + \|\mathbf{C}' \cdot (\mathbf{r}, \mathbf{e}_c, \text{id})\|_\infty \leq q' \cdot \frac{1 + nq'/2 + B_{\text{GPV}} + N}{2}.$$

By choosing $q > q' \cdot \frac{1 + nq'/2 + B_{\text{GPV}} + N}{2}$, and letting

$$\begin{aligned} \bar{\mathbf{A}} &= \begin{pmatrix} \mathbf{C}' & -q' \mathbf{I}_{m_B+1} \end{pmatrix} \in \mathbb{Z}_q^{(m_B+1) \times (n+2m_B+2)}, \\ \bar{\mathbf{x}} &= (\mathbf{r}, \mathbf{e}_c, \text{id}, \mathbf{a}) \in \mathbb{Z}^n \times \mathbb{Z}^{m_B+1} \times \mathbb{Z} \times \mathbb{Z}^{m_B+1}, \end{aligned}$$

we have that $\overline{\mathbf{A}} \cdot \overline{\mathbf{x}} = \mathbf{c} \bmod q$. By using the technique that transforms relation (D.1) to a case of relation (C.1), we can transform $\mathcal{R}_{\text{Enc}}$ to a case of relation (C.1) as well. The final quadratic relation has witness length and number of quadratic constraints approximately $n \log(q') + (m_{\mathbf{B}} + 1) \log(2B_{\text{GPV}}) + \log(N) + (m_{\mathbf{B}} + 1) \log(1 + nq'/2 + B_{\text{GPV}} + N)$.

Proof of LWE Secret. Let $\mathbf{t} \in \mathbb{Z}_{q'}^{m_{\mathbf{M}}}$, $\mathbf{x} \in \mathbb{Z}_{q'}^n$, $\mathbf{e}_{\mathbf{t}} \in \mathbb{Z}^{m_{\mathbf{M}}}$ and $\mathbf{M} \in \mathbb{Z}_{q'}^{m_{\mathbf{M}} \times n}$, we need to prove in ZK the following relation

$$\mathcal{R}_{\text{LWE}}^{(2)} = \{((\mathbf{t}, \mathbf{M}), (\mathbf{x}, \mathbf{e}_{\mathbf{t}})) : \mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e} \bmod q' \wedge \|\mathbf{e}\|_{\infty} \leq B_{\text{LWE}}\}.$$

Write $\mathbf{t} = \mathbf{M} \cdot \mathbf{x} + \mathbf{e} - q' \cdot \mathbf{a}$ for $\mathbf{a} \in \mathbb{Z}^{m_{\mathbf{M}}}$, then

$$q' \cdot \|\mathbf{a}\|_{\infty} \leq \|\mathbf{t}\|_{\infty} + \|\mathbf{M} \cdot \mathbf{x}\|_{\infty} + \|\mathbf{e}\|_{\infty} \leq (q'/2) + (q'/2)^2 + B_{\text{PRF}}.$$

By choosing $q > q' + q'^2 + 2B_{\text{LWE}}$ and letting

$$\begin{aligned} \overline{\mathbf{A}} &= \begin{pmatrix} \mathbf{M} & \mathbf{I}_{m_{\mathbf{M}}} & -q' \mathbf{I}_{m_{\mathbf{M}}} \end{pmatrix} \in \mathbb{Z}_q^{m_{\mathbf{M}} \times (n+2m_{\mathbf{M}})}; \\ \overline{\mathbf{x}} &= (\mathbf{x}, \mathbf{e}_{\mathbf{t}}, \mathbf{a}) \in \mathbb{Z}^n \times \mathbb{Z}^{m_{\mathbf{M}}} \times \mathbb{Z}^{m_{\mathbf{M}}}; \end{aligned}$$

we have that $\overline{\mathbf{A}} \cdot \overline{\mathbf{x}} = \mathbf{t} \bmod q$. By using the technique that transforms relation (D.1) to a case of relation (C.1), we can transform $\mathcal{R}_{\text{LWE}}^{(2)}$ to a case of relation (D.3). The final quadratic relation has witness length and number of quadratic constraints approximately $n \log q' + m_{\mathbf{M}} \log(2B_{\text{LWE}}) + m_{\mathbf{M}} \log(q') \approx 4n \log q'$.

Appendix D.3. Putting Everything Together

To prove relation $\mathcal{R}_{\text{Sign}}$ (Definition 8), each of the statements defining $\mathcal{R}_{\text{Sign}}$ is transformed to a case of quadratic relation as in (C.1):

- Statement proving knowledge of preimage-image under SIS-based function (item (i)), is transformed into $((\mathbf{A}_{\text{SIS}}, \mathbf{y}_{\text{SIS}}, \mathcal{S}_{\text{SIS}}), \mathbf{x}_{\text{SIS}})$, where $\mathbf{A}_{\text{SIS}} \in \mathbb{Z}_q^{n_{\text{SIS}} \times m_{\text{SIS}}}$, $\mathbf{y}_{\text{SIS}} = \mathbf{0}^{n_{\text{SIS}}}$, $\#\mathcal{S}_{\text{SIS}} = m_{\text{SIS}}$ and $\mathbf{x}_{\text{SIS}} \in \{0, 1\}^{m_{\text{SIS}}}$. Here, $n_{\text{SIS}} = n$ and $m_{\text{SIS}} \approx 2n \log q' + 3n \log(n \log q')$;
- Statement proving message-signature pair in the signature scheme by Jeudy *et al.* [44] (item (ii)), is transformed into a case of C.1 (see [44, Appendix F.2]). The final statement is $((\mathbf{A}_{\text{sig}}, \mathbf{y}_{\text{sig}}, \mathcal{S}_{\text{sig}}), \mathbf{x}_{\text{sig}})$, where $\mathbf{A}_{\text{sig}} \in \mathbb{Z}_q^{n_{\text{sig}} \times m_{\text{sig}}}$, $\mathbf{y}_{\text{sig}} \in \mathbb{Z}_q^{n_{\text{sig}}}$ and $\mathbf{x}_{\text{sig}} \in \mathbb{Z}^{m_{\text{sig}}}$. Here, $n_{\text{sig}} = 2n + 1$, $m_{\text{sig}} \approx 1 + \log N + m_1 \log \beta_1 + m_2 \log \beta_2 + m_{\mathbf{B}} \log q' + 2n$; and $\#\mathcal{S}_{\text{sig}} = m_{\text{sig}} - n - 1$;
- Statement proving LWE sample and secret (item (iii)), is transformed into $((\mathbf{A}_{\text{LWE}}^{(1)}, \mathbf{y}_{\text{LWE}}^{(1)}, \mathcal{S}_{\text{LWE}}^{(1)}), \mathbf{x}_{\text{LWE}}^{(1)})$, where $\mathbf{A}_{\text{LWE}}^{(1)} \in \mathbb{Z}_q^{n_{\text{LWE}}^{(1)} \times m_{\text{LWE}}^{(1)}}$, $\mathbf{y}_{\text{LWE}}^{(1)} \in \mathbb{Z}_q^{n_{\text{LWE}}^{(1)}}$, $\#\mathcal{S}_{\text{enc}}^{(1)} = m_{\text{LWE}}^{(1)}$ and $\mathbf{x}_{\text{LWE}}^{(1)} \in \{0, 1\}^{m_{\text{LWE}}^{(1)}}$. Here, $n_{\text{LWE}}^{(1)} = 2m_{\mathbf{B}}$ and $m_{\text{LWE}}^{(1)} \approx 3m_{\mathbf{B}} \log(q') + n \log(q') + m_{\mathbf{B}} \log(2B_{\text{LWE}})$;

- Statement proving plaintext corresponding to a ciphertext of GPV-IBE [45] (item (iv)), is transformed into $((\mathbf{A}_{\text{enc}}, \mathbf{y}_{\text{enc}}, \mathcal{S}_{\text{enc}}), \mathbf{x}_{\text{enc}})$, where $\mathbf{A}_{\text{enc}} \in \mathbb{Z}_q^{n_{\text{enc}} \times m_{\text{enc}}}$, $\mathbf{y}_{\text{enc}} \in \mathbb{Z}_q^{n_{\text{enc}}}$, $\#\mathcal{S}_{\text{enc}} = m_{\text{enc}}$ and $\mathbf{x}_{\text{enc}} \in \{0, 1\}^{m_{\text{enc}}}$. Here, $n_{\text{enc}} = m_{\mathbf{B}} + 1$ and $m_{\text{enc}} \approx n \log(q') + (m_{\mathbf{B}} + 1) \log(2B_{\text{GPV}}) + \log(N) + (m_{\mathbf{B}} + 1) \log(1 + nq'/2 + B_{\text{GPV}} + N)$;
- Statement proving LWE secret (item (v)), is transformed into $((\mathbf{A}_{\text{LWE}}^{(2)}, \mathbf{y}_{\text{LWE}}^{(2)}, \mathcal{S}_{\text{LWE}}^{(2)}), \mathbf{x}_{\text{LWE}}^{(2)})$, where $\mathbf{A}_{\text{LWE}}^{(2)} \in \mathbb{Z}_q^{n_{\text{LWE}}^{(2)} \times m_{\text{LWE}}^{(2)}}$, $\mathbf{y}_{\text{LWE}}^{(2)} \in \mathbb{Z}_q^{n_{\text{LWE}}^{(2)}}$, $\#\mathcal{S}_{\text{LWE}} = m_{\text{LWE}}^{(2)}$ and $\mathbf{x}_{\text{LWE}}^{(2)} \in \{0, 1\}^{m_{\text{LWE}}^{(2)}}$. Here, $n_{\text{LWE}}^{(2)} = m_{\mathbf{M}} = 3n$ and $m_{\text{LWE}}^{(2)} \approx n \log q' + m_{\mathbf{M}} \log(2B_{\text{LWE}}) + m_{\mathbf{M}} \log(q') \approx 4n \log q'$;

From the matrices $\mathbf{A}_{\text{SIS}}, \mathbf{A}_{\text{sig}}, \mathbf{A}_{\text{LWE}}^{(1)}, \mathbf{A}_{\text{enc}}, \mathbf{A}_{\text{LWE}}^{(2)}$; the column vectors $\mathbf{y}_{\text{SIS}}, \mathbf{y}_{\text{sig}}, \mathbf{y}_{\text{LWE}}^{(1)}, \mathbf{y}_{\text{enc}}, \mathbf{y}_{\text{LWE}}^{(2)}$; the sets $\mathcal{S}_{\text{SIS}}, \mathcal{S}_{\text{sig}}, \mathcal{S}_{\text{LWE}}^{(1)}, \mathcal{S}_{\text{enc}}, \mathcal{S}_{\text{LWE}}^{(2)}$ and the witness $\mathbf{x}_{\text{SIS}}, \mathbf{x}_{\text{sig}}, \mathbf{x}_{\text{LWE}}^{(1)}, \mathbf{x}_{\text{enc}}, \mathbf{x}_{\text{LWE}}^{(2)}$; we can form a matrix $\mathbf{A}_{\text{Sign}} \in \mathbb{Z}_q^{n_{\text{Sign}} \times m_{\text{Sign}}}$, a vector $\mathbf{y}_{\text{Sign}} \in \mathbb{Z}_q^{n_{\text{Sign}}}$, a set $\mathcal{S}_{\text{Sign}}$ and a witness $\mathbf{x}_{\text{Sign}} \in \{0, 1\}^{m_{\text{Sign}}}$ such that $\mathbf{A}_{\text{Sign}} \cdot \mathbf{x}_{\text{Sign}} = \mathbf{y}_{\text{Sign}} \bmod q$ and the set $\mathcal{S}_{\text{Sign}} = \{(i, j, k) : i, j, k \in [m_{\text{Sign}}]\}$ enforces the quadratic constraints over the coordinates of $\mathbf{x}_{\text{Sign}}$. In particular, we have

$$n_{\text{Sign}} = n_{\text{SIS}} + n_{\text{sig}} + n_{\text{LWE}}^{(1)} + n_{\text{enc}} + n_{\text{LWE}}^{(2)} = \mathcal{O}(n \log q') = \mathcal{O}(\lambda \log \lambda),$$

and

$$m_{\text{Sign}} = m_{\text{SIS}} + m_{\text{sig}} + m_{\text{LWE}}^{(1)} + m_{\text{enc}} + m_{\text{LWE}}^{(2)} = \mathcal{O}(n(\log q')^2) = \mathcal{O}(\lambda \log^2 \lambda),$$

and the set $\mathcal{S}_{\text{Sign}}$ has size $\ell_{\text{Sign}} \approx m_{\text{Sign}}$. By Proposition 4, the bit size of prover's response in the ZK argument proving $\mathcal{R}_{\text{Sign}}$ is of order $\mathcal{O}(\lambda \log^3 \lambda)$.

Similarly, we can transform statement $\mathcal{R}_{\text{Claim}}$ (Definition 9) by transforming each of the statements defining $\mathcal{R}_{\text{Claim}}$ to a case of (C.1). The final statement has the form $((\mathbf{A}_{\text{Claim}}, \mathbf{y}_{\text{Claim}}, \mathcal{S}_{\text{Claim}}), \mathbf{x}_{\text{Claim}})$, where $\mathbf{A}_{\text{Claim}} \in \mathbb{Z}_q^{n_{\text{Claim}} \times m_{\text{Claim}}}$, $\mathbf{y}_{\text{Claim}} \in \mathbb{Z}_q^{n_{\text{Claim}}}$, $\mathbf{x}_{\text{Claim}} \in \{0, 1\}^{m_{\text{Claim}}}$ and the set $\mathcal{S}_{\text{Claim}}$ enforces the binary constraints over the coordinates of $\mathbf{x}_{\text{Claim}}$. In particular, we have

$$n_{\text{Claim}} = n_{\text{SIS}} + n_{\text{LWE}}^{(2)} = n + 3n = 4n = \mathcal{O}(\lambda),$$

and

$$m_{\text{Claim}} = m_{\text{SIS}} + m_{\text{LWE}}^{(2)} \approx n + 4n \log q' = \mathcal{O}(n \log q') = \mathcal{O}(\lambda \log \lambda),$$

and $\mathcal{S}_{\text{Claim}}$ has size $\ell_{\text{Claim}} = 4n = \mathcal{O}(\lambda)$. By Proposition 4, the bit size of prover's response in the ZK argument proving $\mathcal{R}_{\text{Claim}}$ is of order $\mathcal{O}(\lambda \log^2 \lambda)$.

From $(n_{\text{Sign}}, m_{\text{Sign}}, \ell_{\text{Sign}})$ and $(n_{\text{Claim}}, m_{\text{Claim}}, \ell_{\text{Claim}})$, one sets up the common reference string crs in the ZK argument system by choosing an auxiliary string commitment schemes and public parameters of BDLP commitment scheme [62]. To prove $\mathcal{R}_{\text{Sign}}$ and $\mathcal{R}_{\text{Claim}}$, signers run the ZK protocol with generalized Unruh transformations using the respective hash functions.