Accountable Tracing Signatures from Lattices

San Ling, Khoa Nguyen, Huaxiong Wang, Yanhong Xu

Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore.
{lingsan,khoantt,hxwang,xu0014ng}@ntu.edu.sg

Abstract. Group signatures allow users of a group to sign messages anonymously in the name of the group, while incorporating a tracing mechanism to revoke anonymity and identify the signer of any message. Since its introduction by Chaum and van Heyst (EUROCRYPT 1991), numerous proposals have been put forward, yielding various improvements on security, efficiency and functionality. However, a drawback of traditional group signatures is that the opening authority is given too much power, i.e., he can indiscriminately revoke anonymity and there is no mechanism to keep him accountable. To overcome this problem, Kohlweiss and Miers (PoPET 2015) introduced the notion of accountable tracing signatures (ATS) - an enhanced group signature variant in which the opening authority is kept accountable for his actions. Kohlweiss and Miers demonstrated a generic construction of ATS and put forward a concrete instantiation based on number-theoretic assumptions. To the best of our knowledge, no other ATS scheme has been known, and the problem of instantiating ATS under post-quantum assumptions, e.g., lattices, remains open to date.

In this work, we provide the first lattice-based accountable tracing signature scheme. The scheme satisfies the security requirements suggested by Kohlweiss and Miers, assuming the hardness of the Ring Short Integer Solution (RSIS) and the Ring Learning With Errors (RLWE) problems. At the heart of our construction are a lattice-based key-oblivious encryption scheme and a zero-knowledge argument system allowing to prove that a given ciphertext is a valid RLWE encryption under some hidden yet certified key. These technical building blocks may be of independent interest, e.g., they can be useful for the design of other lattice-based privacy-preserving protocols.

1 Introduction

Group signature is a fundamental cryptographic primitive introduced by Chaum and van Heyst [13]. It allows members of a group to anonymously sign messages on behalf of the group, but to prevent abuse of anonymity, there is an opening authority (OA) who can identify the signer of any message. While such a tracing mechanism is necessary to ensure user accountability, it grants too much power to the opening authority. Indeed, in traditional models of group signatures,

e.g., [2,23,7,3,24,54,8], the OA can break users' anonymity whenever he wants, and we do not have any method to verify whether this trust is well placed or not.

One existing attempt to restrict the OA's power is the proposal of group signatures with message-dependent opening (MDO) [53], in which the OA can only identify the signers of messages admitted by an additional authority named admitter. However, this solution is still unsatisfactory. Once the OA has obtained admission to open a specific message, he can identify all the users, including some innocent ones, who have ever issued signatures on this specific message. Furthermore, by colluding with the admitter, the OA again is able to open all signatures.

To tackle the discussed above problem, Kohlweiss and Miers [25] put forward the notion of accountable tracing signatures (ATS), which is an enhanced variant of group signatures that has an additional mechanism to make the OA accountable. In an ATS scheme, the role of the OA is incorporated into that of the group manager (GM), and there are two kinds of group users: traceable ones and non-traceable ones. Traceable users are treated as in traditional group signatures, i.e., their anonymity can be broken by the OA/GM. Meanwhile, it is infeasible for anyone, including the OA/GM, to trace signatures generated by non-traceable users. When a user joins the group, the OA/GM first has to determine whether this user is traceable and then he issues a corresponding (traceable/nontraceable) certificate to the user. In a later phase, the OA/GM reveals which user he deems traceable using an "accounting" algorithm, yielding an intriguing method to enforce his accountability.

As an example, let us consider the surveillance controls of a building, which is implemented using an ATS scheme. On the one hand, the customers in this building would like to have their privacy protected as much as possible. On the other hand, the police who are conducting security check in this building would like to know as much as they can. To balance the interests of these two parties, the police can in advance narrow down some suspects and asks the OA/GM to make these suspected users traceable and the remaining non-suspected users non-traceable. To check whether the suspects entered the building, the police can ask the OA/GM to open all signatures that were used for authentication at the entrance. Since only the suspects are traceable, the group manager can only identify them if they indeed entered this building. However, if a standard group signature scheme (e.g., [1,2,6,3]) were used, then the privacy of innocent users would be seriously violated. In this situation, one might think that a traceable signature scheme, as suggested by Kiayias, Tsiounis and Yung [23], would work. By requesting a user-specific trapdoor from the OA/GM, the police can trace all the signatures created by the suspects. However, this only achieves privacy of innocent users against the *police*, but not against the *group authorities*. In fact, in a traceable signature scheme, the OA/GM has the full power to identify the signers of all signatures and hence can violate the privacy of all users without being detected. In contrast, if an ATS scheme is used, then the OA/GM must later reveal which user he chose to be traceable, thus enabling his accountability.

In [25], besides demonstrating the feasibility of ATS under generic assumptions, Kohlweiss and Miers also presented an instantiation based on number-theoretic assumptions, which remains the only known concrete ATS construction to date. This scheme, however, is vulnerable against quantum computers due to Shor's algorithm [55]. For the sake of not putting all eggs in one basket, it is therefore tempting to build schemes based on post-quantum foundations. In this paper, we investigate the design of accountable tracing signatures based on lattice assumptions, which are currently among the most viable foundations for post-quantum cryptography. Let us now take a look at the closely related and recently active topic of lattice-based group signatures.

Lattice-based group signatures. The first lattice-based group signature scheme was introduced by Gordon, Katz and Vaikuntanathan in 2010 [20]. Subsequently, numerous schemes offering improvements in terms of security and efficiency have been proposed [12,26,34,48,30,28,9,51]. Nevertheless, regarding the supports of advanced functionalities, lattice-based group signatures are still way behind their number-theoretic-based counterparts. Indeed, there have been known only a few lattice-based schemes [32,31,28,35,36] that depart from the BMW model [2] - which deals solely with static groups and which may be too inflexible to be considered for a wide range of real-life applications. In particular, although there was an attempt [31] to restrict the power of the OA in the MDO sense, the problem of making the OA accountable in the context of lattice-based group signatures is still open. This somewhat unsatisfactory state-of-affairs motivates our search for a lattice-based instantiation of ATS. As we will discuss below, the technical road towards our goal is not straightforward: there are challenges and missing building blocks along the way.

Our Results and Techniques. In this paper, we introduce the first lattice-based accountable tracing signature scheme. The scheme satisfies the security requirements suggested by Kohlweiss and Miers [25], assuming the hardness of the Ring Short Integer Solution (RSIS) problem and the Ring Learning With Errors (RLWE) problem. As all other known lattice-based group signatures, the security of our scheme is analyzed in the random oracle model. For a security parameter λ, our ATS scheme features group public key size and user secret key size $\widetilde{\mathcal{O}}(\lambda)$. However, the accountability of the OA/GM comes at a price: the signature size is of order $\widetilde{\mathcal{O}}(\lambda^2)$ compared with $\widetilde{\mathcal{O}}(\lambda)$ in a recent scheme by Ling et al. [36].

Let us now give an overview of our techniques. First, we recall that in an ordinary group signature scheme [2,3], to enable traceability, the user is supposed to encrypt his identifying information and prove the well-formedness of the resulting ciphertext. In an ATS scheme, however, not all users are traceable. We thus would need a mechanism to distinguish between traceable users and non-traceable ones. A possible method is to let traceable users encrypt their identities under a public key (pk) such that only the OA/GM knows the underlying secret key (sk), while for non-traceable users, no one knows the secret key. However, there seems to be no incentive for users to deliberately make themselves traceable. We hence should think of a way to choose traceable users obliviously. An

interesting approach is to randomize pk to a new public key epk so that it is infeasible to decide how these keys are related without the knowledge of the secret key and the used randomness. More specifically, when a user joins the group, the OA/GM first randomizes pk to epk and sends the latter to the user together with a certificate. The difference between traceable users and non-traceable ones lies in whether OA/GM knows the underlying secret key. Thanks to the obliviousness property of the randomization, the users are unaware of whether they are traceable. Then, when signing messages, the user encrypts his identity using his own randomized key epk (note that this "public key" should be kept secret) and proves the well-formedness of the ciphertext. Several questions regarding this approach then arise. What special kind of encryption scheme should we use? How to randomize the public key in order to get the desirable obliviousness? More importantly, how could the user prove the honest execution of encryption if the underlying encryption key is secret?

To address the first two questions, Kohlweiss and Miers [25] proposed the notion of key-oblivious encryption (KOE) - a public-key encryption scheme in which one can randomize public keys in an oblivious manner. Kohlweiss and Miers showed that a KOE scheme can be built from a key-private homomorphic public-key encryption scheme. They then gave an explicit construction based on the ElGamal cryptosystem [18], where epk is obtained by multiplying pk by a ciphertext of 1. When adapting this idea into the lattice setting, however, one has to be careful. In fact, we observe that an implicit condition for the underlying key-private public-key encryption scheme is that its public key and ciphertext should have *the same algebraic form*[1], which is often not the case for the schemes in the lattice setting, e.g., [52,19]. Furthermore, lattice-based encryption schemes from the Learning with Errors (LWE) problem or its ring version RLWE often involve noise terms that grow quickly when one performs homomorphic operations over ciphertexts. Fortunately, we could identify a suitable candidate: the RLWE-based encryption scheme proposed by Lyubashevsky, Peiker and Regev (LPR) [43], for which both the public key and the ciphertext consist of a pair of ring elements. Setting the parameters carefully to control the noise growth in LPR, we are able to adapt the blueprint of [25] into the lattice setting and obtain a lattice-based KOE scheme.

To tackle the third question, we need a zero-knowledge (ZK) protocol for proving well-formedness of the ciphertext under a hidden encryption key, which is quite challenging to build in the RLWE setting. Existing ZK protocols from lattices belong to two main families. One line of research [37,38,4,5,41,44] designed very elegant approximate ZK proofs for (R)LWE and (R)SIS relations by employing rejection sampling techniques. While these proofs are quite efficient and compact, they only handle linear relations. In other words, they can only prove knowledge of a short vector $\mathbf{x}$ satisfying $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} \bmod q$, for *public* $\mathbf{A}$ and public $\mathbf{y}$. This seems insufficient for our purpose. Another line of research [33,34,14,30,29,36] developed decomposition/ extension/permutation

[1] This condition is needed so that epk can be computed as pk $\cdot$ enc(1) (multiplicative homomorphic) or pk $+$ enc(0) (additive homomorphic).

techniques that operate in Stern's framework [57]. Although Stern-like protocols are less practical than those in the first family, they are much more versatile and can even deal with quadratic relations [29]. More precisely, as demonstrated by Libert et al. [29] one can employ Stern-like techniques to prove knowledge of *secret-and-certified* $\mathbf{A}$ together with short secret vector $\mathbf{x}$ satisfying $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} \bmod q$. Thus, Libert et al.'s work appears to be the "right" stepping stone for our case. However, in [29], quadratic relations were considered only in the setting of general lattices, while here we have to deal with the ring setting, for which the multiplication operation is harder to express, capture and prove in zero-knowledge. Nevertheless we manage to adapt their techniques into the ring lattices and obtain the desired technical building block.

As discussed so far, we have identified the necessary ingredients - the LPR encryption scheme and Stern-like ZK protocols - for upgrading a lattice-based ordinary group signature to a lattice-based accountable tracing signature. Next, we need to find a lattice-based ordinary group signature scheme that is compatible with the those ingredients. To this end, we work with Ling et al.'s scheme [36], that also employs the LPR system for its tracing layer and Stern-like techniques for proving knowledge of a valid user certificate (which is a Ducas-Micciancio signature [15,16] based on the hardness of the Ring Short Integer Solution (RSIS) problem). We note that the scheme from [36] achieves constant-size signatures, which means that the signature size is independent of the number of users. As a by-product, our signatures are also constant-size (although our constant is larger, due to the treatment of quadratic relations).

A remaining aspect is how to enable the accountability of the OA/GM. To this end, we let the latter reveal the choice (either traceable or non-traceable) for a given user together with the randomness used to obtain the randomized public key. The user then checks whether his epk was computed as claimed. However, the OA/GM may claim a traceable user to be non-traceable by giving away malicious randomness and accusing that the user had changed epk by himself. To ensure non-repudiation, OA/GM is required to sign epk and the users' identifying information when registering the user into the group. This mechanism in fact also prevents dishonest users from choosing non-traceable epk by themselves.

The obtained ATS scheme is then proven secure in the random oracle model under the RSIS and RLWE assumptions, according to the security requirements put forward by Kohlweiss and Miers [25]. On the efficiency front, as all known lattice-based group signatures with advanced functionalities, our scheme is still far from being practical. We, however, hope that our result will inspire more efficient constructions in the near future.

Organization. In Section 2, we recall some background materials. In Section 3, we describe our key-oblivious encryption scheme from lattice assumptions. Our accountable tracing signature scheme is presented in Section 5.

2 Background

NOTATIONS. For a positive integer n, define the set $\{1, 2, \ldots, n\}$ as $[n]$, the set $\{0, 1, \ldots, n\}$ as $[0, n]$, and the set containing all the integers from $-n$ to n as $[-n, n]$. Denote the set of all positive integers as $\mathbb{Z}^+$. If S is a finite set, then $x \xleftarrow{\$} S$ means that x is chosen uniformly at random from S. Let $\mathbf{a} \in \mathbb{R}^{m_1}$ and $\mathbf{b} \in \mathbb{R}^{m_2}$ be two vectors for positive integers m_1, m_2. Denote $(\mathbf{a}\|\mathbf{b}) \in \mathbb{R}^{m_1+m_2}$, instead of $(\mathbf{a}^\top, \mathbf{b}^\top)^\top$, as the concatenation of these two vectors.

2.1 Rings, RSIS and RLWE

Let $q \geq 3$ be a positive integer and let $\mathbb{Z}_q = [-\frac{q-1}{2}, \frac{q-1}{2}]$. In this work, let us consider rings $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = (R/qR)$, where n is a power of 2.

Let τ be the coefficient embedding $\tau : R_q \to \mathbb{Z}_q^n$ that maps a ring element $v = v_0 + v_1 \cdot X + \ldots + v_{n-1} \cdot X^{n-1} \in R_q$ to a vector $\tau(v) = (v_0, v_1, \ldots, v_{n-1})^\top$ over $\mathbb{Z}_q^n$. Define the ring homomorphism $\mathsf{rot} : R_q \to \mathbb{Z}_q^{n \times n}$that maps a ring element $a \in R_q$ to a matrix $\mathsf{rot}(a) = \big[\tau(a) \mid \tau(a \cdot X) \mid \cdots \mid \tau(a \cdot X^{n-1})\big]$ over $\mathbb{Z}_q^{n \times n}$ (see, e.g., [45,58]). Using these two functions, the element product $y = a \cdot v$ over R_q can be interpreted as the matrix-vector multiplication $\tau(y) = \mathsf{rot}(a) \cdot \tau(v)$ over $\mathbb{Z}_q$.

When working with vectors and matrices over R_q, we generalize the notations τ and rot in the following way. For a vector $\mathbf{v} = (v_1, \ldots, v_m)^\top \in R_q^m$, define $\tau(\mathbf{v}) = (\tau(v_1)\| \cdots \|\tau(v_m)) \in \mathbb{Z}_q^{mn}$. For a matrix $\mathbf{A} = [a_1 \mid \cdots \mid a_m] \in R_q^{1 \times m}$, define $\mathsf{rot}(A)$ to be the matrix

$$\mathsf{rot}(\mathbf{A}) = \big[\mathsf{rot}(a_1) \mid \cdots \mid \mathsf{rot}(a_m)\big] \in \mathbb{Z}_q^{n \times mn}.$$

Using the generalized notations, we can interpret $y = \mathbf{A} \cdot \mathbf{v}$ over R_q as matrix-vector multiplication $\tau(y) = \mathsf{rot}(\mathbf{A}) \cdot \tau(\mathbf{v})$ over $\mathbb{Z}_q$.

For $a = a_0 + a_1 \cdot X + \ldots + a_{n-1} \cdot X^{N-1} \in R$, we define $\|a\|_\infty = \max_i(|a_i|)$. Similarly, for vector $\mathbf{b} = (b_1, \ldots, b_\mathtt{m})^\top \in R^\mathtt{m}$, we define $\|\mathbf{b}\|_\infty = \max_j(\|b_j\|_\infty)$.

We now recall the average-case problems RSIS and RLWE associated with the rings R, R_q, as well as their hardness results.

Definition 1 ([39,50,40]). *Given a uniform matrix $\mathbf{A} = [a_1|a_2|\cdots|a_m]$ over $R_q^{1 \times m}$, the $\mathsf{RSIS}^\infty_{n,m,q,\beta}$ problem asks to find a ring vector $\mathbf{b} = (b_1, b_2, \ldots, b_m)^\top$ over R^m such that $\mathbf{A} \cdot \mathbf{b} = a_1 \cdot b_1 + a_2 \cdot b_2 + \cdots + a_m \cdot b_m = 0$ over R_q and $0 < \|\mathbf{b}\|_\infty \leq \beta$.*

For polynomial bounded m, β and $q \geq \beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$, it was proven that the $\mathsf{RSIS}^\infty_{n,m,q,\beta}$ problem is no easier than the SIVP_γ problem in any ideal in the ring R, where $\gamma = \beta \cdot \widetilde{\mathcal{O}}(\sqrt{nm})$ (see [39,50,27]).

Definition 2 ([42,56,43]). *For positive integers $n, m, q \geq 2$ and a probability distribution χ over the ring R, define a distribution $A_{s,\chi}$ over $R_q \times R_q$ for $s \xleftarrow{\$} R_q$ in the following way: it first samples a uniformly random element $a \in R_q$, an*

error element $e \hookleftarrow \chi$*, and then outputs* $(a, a \cdot s + e)$*. The target of the* $\mathsf{RLWE}_{n,m,q,\chi}$ *problem is to distinguish* m *samples chosen from a uniform distribution over* $R_q \times R_q$ *and* m *samples chosen from the distribution* $A_{s,\chi}$ *for* $s \xleftarrow{\$} R_q$*.*

Let $q \geq 2$ and $B = \widetilde{\mathcal{O}}(\sqrt{n})$ be positive integers. χ is a distribution over R which efficiently outputs samples $e \in R$ with $\|e\|_\infty \leq B$ with overwhelming probability in n. Then there is a quantum reduction from the $\mathsf{RLWE}_{n,m,q,\chi}$ problem to the SIVP_γ problem and the SVP_γ problem in any ideal in the ring R, where $\gamma = \widetilde{\mathcal{O}}(\sqrt{n} \cdot q/B)$ (see [42,10,27,49]). It is shown that the hardness of the RLWE problem is preserved when the secret s is sampled from the error distribution χ (see [42,10]).

2.2 Decompositions

We now recall the integer decomposition technique from [33]. For any positive integer B, let $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor$, for any $j \in [\delta_B]$. It is then verifiable that $\sum_{j=1}^{\delta_B} B_j = B$. In addition, for any integer $a \in [0, B]$, one can decompose a into a vector of the form $\mathsf{idec}_B(a) = (a^{(1)}, a^{(2)}, \ldots, a^{(\delta_B)})^\top \in \{0,1\}^{\delta_B}$, satisfying $(B_1, B_2, \ldots, B_{\delta_B}) \cdot \mathsf{idec}_B(a) = a$. The procedure of the decomposition is presented below in a deterministic manner.

1. $a' := a$
2. For $j = 1$ to δ_B do:
   (i) If $a' \geq B_j$ then $a^{(j)} := 1$, else $a^{(j)} := 0$;
   (ii) $a' := a' - B_j \cdot a^{(j)}$.
3. Output $\mathsf{idec}_B(a) = (a^{(1)}, \ldots, a^{(\delta_B)})^\top$.

In [36], the above decomposition procedure is also utilized to deal with polynomials in the ring R_q. Specifically, for $B \in [1, \frac{q-1}{2}]$, define the injective function rdec_B that maps $a \in R_q$ with $\|a\|_\infty \leq B$ to $\mathbf{a} \in R^{\delta_B}$ with $\|\mathbf{a}\|_\infty \leq 1$, which works as follows.

1. Let $\tau(a) = (a_0, \ldots, a_{n-1})^\top$. For each i, let $\sigma(a_i) = 0$ if $a_i = 0$; $\sigma(a_i) = -1$ if $a_i < 0$; and $\sigma(a_i) = 1$ if $a_i > 0$.
2. $\forall i$, compute $\mathbf{w}_i = \sigma(a_i) \cdot \mathsf{idec}_B(|a_i|) = (w_{i,1}, \ldots, w_{i,\delta_B})^\top \in \{-1, 0, 1\}^{\delta_B}$.
3. Form the vector $\mathbf{w} = (\mathbf{w}_0 \| \ldots \| \mathbf{w}_{n-1}) \in \{-1, 0, 1\}^{n\delta_B}$, and let $\mathbf{a} \in R^{\delta_B}$ be the vector such that $\tau(\mathbf{a}) = \mathbf{w}$.
4. Output $\mathsf{rdec}_B(a) = \mathbf{a}$.

To deal with ring vectors of dimension $m \in \mathbb{Z}^+$ and of infinity bound $B \in \mathbb{Z}^+$, we generalize the notion $\mathsf{rdec}_B(\mathbf{v})$ in the following way: it maps a ring vector $\mathbf{v} = (v_1, \ldots, v_m)^\top \in R_q^m$ such that $\|\mathbf{v}\|_\infty \leq B$ to a vector $\mathsf{rdec}_B(\mathbf{v}) = \big(\mathsf{rdec}_B(v_1) \| \ldots \| \mathsf{rdec}_B(v_m)\big) \in R^{m\delta_B}$, whose coefficients are in the set $\{-1, 0, 1\}$.

Now, $\forall m, B \in \mathbb{Z}^+$, we define matrices $\mathbf{H}_B \in \mathbb{Z}^{n \times n\delta_B}$ and $\mathbf{H}_{m,B} \in \mathbb{Z}^{nm \times nm\delta_B}$ as

$$\mathbf{H}_B = \begin{bmatrix} B_1 \dots B_{\delta_B} & & \\ & \ddots & \\ & & B_1 \dots B_{\delta_B} \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_{m,B} = \begin{bmatrix} \mathbf{H}_B & & \\ & \ddots & \\ & & \mathbf{H}_B \end{bmatrix}.$$

Then we have

$$\tau(a) = \mathbf{H}_B \cdot \tau(\mathsf{rdec}_B(a)) \bmod q \quad \text{and} \quad \tau(\mathbf{v}) = \mathbf{H}_{m,B} \cdot \tau(\mathsf{rdec}_B(\mathbf{v})).$$

For simplicity reason, when $B = \frac{q-1}{2}$, we will use the notation rdec instead of $\mathsf{rdec}_{\frac{q-1}{2}}$, and $\mathbf{H}$ instead of $\mathbf{H}_{\frac{q-1}{2}}$.

2.3 A Variant of the Ducas-Micciancio Signature scheme

We recall the stateful and adaptively secure version of Ducas-Micciancio signature scheme [15,16], which is used to enroll new users in our construction.

Following [15,16], throughout this work, for any real constants $c > 1$ and $\alpha_0 \geq \frac{1}{c-1}$, define a series of sets $\mathcal{T}_j = \{0,1\}^{c_j}$ of lengths $c_j = \lfloor \alpha_0 c^j \rfloor$ for $j \in [d]$, where $d \geq \log_c(\omega(\log n))$. For each tag $t = (t_0, t_1, \dots, t_{c_j})^\top \in \mathcal{T}_j$ for $j \in [d]$, associate it with a ring element $t(X) = \sum_{k=0}^{c_j} t_k \cdot X^k \in R_q$. Let $c_0 = 0$ and then define $t_{[i]}(X) = \sum_{k=c_{i-1}}^{c_i - 1} t_k \cdot X^k$ and $t_{[i]} = (t_{c_{i-1}}, \dots, t_{c_i - 1})^\top$ for $i \in [j]$. Then one can check $t = (t_{[1]} \| t_{[2]} \| \cdots \| t_{[j]})$ and $t(X) = \sum_{i=1}^{j} t_{[i]}(X)$.

This variant works with the following parameters.

- Let n, m, q, k be some positive integers such that $n \geq 4$ is a power of 2, $m \geq 2\lceil \log q \rceil + 2$, and $q = 3^k$. Define the rings $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = R/qR$.
- Let the message dimension be $m_s = \mathrm{poly}(n)$. Also, let $\ell = \lfloor \log \frac{q-1}{2} \rfloor + 1$, and $\overline{m} = m + k$ and $\overline{m}_s = m_s \cdot \ell$.
- Let integer $\beta = \widetilde{\mathcal{O}}(n)$ and integer d and sequence $c_0, \dots, c_d$ be as above.
- Let $S \in \mathbb{Z}$ be a state that is 0 initially.

The public verification key consists of the following:

$$\mathbf{A}, \mathbf{F}_0 \in R_q^{1 \times \overline{m}}; \quad \mathbf{A}_{[0]}, \dots, \mathbf{A}_{[d]} \in R_q^{1 \times k}; \quad \mathbf{F} \in R_q^{1 \times \ell}; \quad \mathbf{F}_1 \in R_q^{1 \times \overline{m}_s}; \quad u \in R_q$$

while the secret signing key is a Micciancio-Peikert [46] trapdoor matrix $\mathbf{R} \in R_q^{m \times k}$.

When signing a message $\mathfrak{m} \in R_q^{m_s}$, the signer first computes $\overline{\mathfrak{m}} = \mathsf{rdec}(\mathfrak{m}) \in R^{\overline{m}_s}$, whose coefficients are in the set $\{-1, 0, 1\}$. He then performs the following steps.

- Set the tag $t = (t_0, t_1 \dots, t_{c_d - 1})^\top \in \mathcal{T}_d$, where $S = \sum_{j=0}^{c_d - 1} 2^j \cdot t_j$, and compute $\mathbf{A}_t = [\mathbf{A} | \mathbf{A}_{[0]} + \sum_{i=1}^{d} t_{[i]} \mathbf{A}_{[i]}] \in R_q^{1 \times (\overline{m} + k)}$. Update S to $S + 1$.

– Choose $\mathbf{r} \in R^{\overline{m}}$ with $\|\mathbf{r}\|_\infty \leq \beta$.
– Let $y = \mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \overline{\mathfrak{m}} \in R_q$ and $u_p = \mathbf{F} \cdot \mathsf{rdec}(y) + u \in R_q$.
– Employing the trapdoor matrix $\mathbf{R}$, produce a ring vector $\mathbf{v} \in R^{\overline{m}+k}$ with $\mathbf{A}_t \cdot \mathbf{v} = u_p$ over the ring R_q and $\|\mathbf{v}\|_\infty \leq \beta$.
– Return the tuple $(t, \mathbf{r}, \mathbf{v})$ as a signature for the message $\mathfrak{m}$.

To check the validity of the tuple $(t, \mathbf{r}, \mathbf{v})$ with respect to message $\mathfrak{m} \in R_q^{m_s}$, the verifier first computes the matrix $\mathbf{A}_t$ as above and verifies the following conditions:

$$\begin{cases} \mathbf{A}_t \cdot \mathbf{v} = \mathbf{F} \cdot \mathsf{rdec}(\mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \mathsf{rdec}(\mathfrak{m})) + u, \\ \|\mathbf{r}\|_\infty \leq \beta, \quad \|\mathbf{v}\|_\infty \leq \beta. \end{cases}$$

He outputs 1 if all these three conditions hold and 0 otherwise.

Lemma 1 ([15,16]). *Given at most polynomially bounded number of signature queries, the above variant is existentially unforgeable against adaptive chosen message attacks assuming the hardness of the* $\mathsf{RSIS}_{n,\overline{m},q,\widetilde{\mathcal{O}}(n^2)}$ *problem.*

2.4 Zero-Knowledge Argument of Knowledge

We will work with statistical zero-knowledge argument systems, namely, interactive protocols where the ZK property holds against *any* cheating verifier, while the soundness property only holds against *computationally bounded* cheating provers. More formally, let the set of statements-witnesses $\mathrm{R} = \{(y, w)\} \in \{0,1\}^* \times \{0,1\}^*$ be an NP relation. A two-party game $\langle \mathcal{P}, \mathcal{V} \rangle$ is called an interactive argument system for the relation R with soundness error e if the following two conditions hold:

– Completeness. If $(y, w) \in \mathrm{R}$ then $\Pr\big[\langle \mathcal{P}(y, w), \mathcal{V}(y) \rangle = 1\big] = 1$.
– Soundness. If $(y, w) \notin \mathrm{R}$, then $\forall$ PPT $\widehat{\mathcal{P}}$: $\Pr[\langle \widehat{\mathcal{P}}(y, w), \mathcal{V}(y) \rangle = 1] \leq e$.

An argument system is called statistical ZK if for any $\widehat{\mathcal{V}}(y)$, there exists a PPT simulator $\mathcal{S}(y)$ having oracle access to $\widehat{\mathcal{V}}(y)$ and producing a simulated transcript that is statistically close to the one of the real interaction between $\mathcal{P}(y, w)$ and $\widehat{\mathcal{V}}(y)$. A related notion is argument of knowledge, which, for three-move protocols (commitment-challenge-response), requires the existence of a PPT extractor taking as input a set of valid transcripts with respect to all possible values of the "challenge" to the same "commitment" and outputting w' such that $(y, w') \in \mathrm{R}$.

The statistical zero-knowledge arguments of knowledge (ZKAoK) presented in this work are Stern-like [57] protocols. In particular, they are Σ-protocols in the generalized sense defined in [21,4] (where 3 valid transcripts are needed for extraction, instead of just 2). Stern's protocol was originally proposed in the context of code-based cryptography, and was later adapted into the lattice setting by Kawachi et al. [22]. Subsequently, it was empowered by Ling et al. [33]

to handle the matrix-vector relations where the secret vectors are of small infinity norm, and further developed to design various lattice-based schemes. Libert et al. [28] put forward an abstraction of Stern's protocol to capture a wider range of lattice-based relations. Now let us recall it.

An Abstraction of Stern's Protocol. Let integers q, K, L be positive such that $L \geq K$ and $q \geq 2$, and $\mathsf{VALID} \subset \{-1, 0, 1\}^L$. Given a finite set $\mathcal{S}$, associate every $\eta \in \mathcal{S}$ with a permutation Γ_η of L elements such that the following conditions hold:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\eta(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \eta \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\eta(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{1}$$

Our target is to construct a statistical ZKAoK for the abstract relation $\mathrm{R}_{\mathrm{abstract}}$ of the following form:

$$\mathrm{R}_{\mathrm{abstract}} = \left\{ (\mathbf{M}, \mathbf{u}), \mathbf{w} \in \mathbb{Z}_q^{K \times L} \times \mathbb{Z}_q^K \times \mathsf{VALID} : \mathbf{M} \cdot \mathbf{w} = \mathbf{u} \bmod q. \right\}$$

To obtain the desired ZKAoK protocol, one has to prove that $\mathbf{w} \in \mathsf{VALID}$ and $\mathbf{w}$ satisfies the linear equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{u} \bmod q$. To prove $\mathbf{w} \in \mathsf{VALID}$ in a zero-knowledge manner, the prover chooses $\eta \xleftarrow{\$} \mathcal{S}$ and allows the verifier to check $\Gamma_\eta(\mathbf{w}) \in \mathsf{VALID}$. According to the first condition in (1), the verifier should be convinced that $\mathbf{w}$ is indeed from the set VALID. At the same time, the verifier cannot learn any extra information about $\mathbf{w}$ due to the second condition in (1). Furthermore, to prove in ZK that the linear equation holds, the prover first chooses $\mathbf{r}_w \xleftarrow{\$} \mathbb{Z}_q^L$ as a masking vector and then shows the verifier that the equation $\mathbf{M} \cdot (\mathbf{w} + \mathbf{r}_w) = \mathbf{M} \cdot \mathbf{r}_w + \mathbf{u} \bmod q$ holds.

In Figure 1, we describe in details the interaction between two PPT algorithms prover $\mathcal{P}$ and verifier $\mathcal{V}$. The system utilizes a statistically hiding and computationally binding string commitment scheme COM (e.g., the RSIS-based scheme from [22]).

Theorem 1 ([28]). *Let* COM *be a statistically hiding and computationally binding string commitment scheme. Then the interactive protocol depicted in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(L \log q)$*. Specifically:*

- *There exists a polynomial-time simulator that on input* $(\mathbf{M}, \mathbf{u})$*, with probability* $2/3$ *it outputs an accepted transcript that is within statistical distance from the one produced by an honest prover who knows the witness.*
- *There exists a polynomial-time algorithm that, takes as inputs* $(\mathbf{M}, \mathbf{u})$ *and three accepting transcripts on* $(\mathbf{M}, \mathbf{u})$*,* $(\mathrm{CMT}, 1, \mathrm{RSP}_1)$*,* $(\mathrm{CMT}, 2, \mathrm{RSP}_2)$*, and* $(\mathrm{CMT}, 3, \mathrm{RSP}_3)$*, outputs* $\mathbf{w}' \in \mathsf{VALID}$ *such that* $\mathbf{M} \cdot \mathbf{w}' = \mathbf{u} \bmod q$*.*

The details of the proof appeared in [28] and are omitted here.

1. **Commitment:** Prover chooses $\mathbf{r}_w \xleftarrow{\$} \mathbb{Z}_q^L$, $\eta \xleftarrow{\$} \mathcal{S}$ and randomness ρ_1, ρ_2, ρ_3 for COM. Then he sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to the verifier, where

$$C_1 = \mathsf{COM}(\eta, \mathbf{M} \cdot \mathbf{r}_w \bmod q; \rho_1), \;\; C_2 = \mathsf{COM}(\Gamma_\eta(\mathbf{r}_w); \rho_2),$$
$$C_3 = \mathsf{COM}(\Gamma_\eta(\mathbf{w} + \mathbf{r}_w \bmod q); \rho_3).$$

2. **Challenge:** $\mathcal{V}$ sends back a challenge $Ch \xleftarrow{\$} \{1, 2, 3\}$ to $\mathcal{P}$.
3. **Response:** According to the choice of Ch, $\mathcal{P}$ sends back RSP computed in the following way:
   - $Ch = 1$: Let $\mathbf{t}_w = \Gamma_\eta(\mathbf{w})$, $\mathbf{t}_r = \Gamma_\eta(\mathbf{r}_w)$, and $\mathrm{RSP} = (\mathbf{t}_w, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\eta_2 = \eta$, $\mathbf{w}_2 = \mathbf{w} + \mathbf{r}_w \bmod q$, and $\mathrm{RSP} = (\eta_2, \mathbf{w}_2, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\eta_3 = \eta$, $\mathbf{w}_3 = \mathbf{r}_w$, and $\mathrm{RSP} = (\eta_3, \mathbf{w}_3, \rho_1, \rho_2)$.

Verification: When receiving RSP from $\mathcal{P}$, $\mathcal{V}$ performs as follows:

- $Ch = 1$: Check that $\mathbf{t}_w \in \mathsf{VALID}$, $C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2)$, $C_3 = \mathsf{COM}(\mathbf{t}_w + \mathbf{t}_r \bmod q; \rho_3)$.
- $Ch = 2$: Check that $C_1 = \mathsf{COM}(\eta_2, \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{u} \bmod q; \rho_1)$, $C_3 = \mathsf{COM}(\Gamma_{\eta_2}(\mathbf{w}_2); \rho_3)$.
- $Ch = 3$: Check that $C_1 = \mathsf{COM}(\eta_3, \mathbf{M} \cdot \mathbf{w}_3; \rho_1)$, $C_2 = \mathsf{COM}(\Gamma_{\eta_3}(\mathbf{w}_3); \rho_2)$.

In each case, $\mathcal{V}$ returns 1 if and only if all the conditions hold.

Fig. 1: Stern-like ZKAoK for the relation $\mathrm{R}_{\mathrm{abstract}}$.

2.5 The Refined Permuting Techniques by Ling et al.

We next recall the permuting techniques recently suggested by Ling et al. [36], which will be used throughout this paper.

Proving that $z \in \{-1, 0, 1\}$. Let b an integer. Denote the integer $b' \in \{-1, 0, 1\}$ with $b' = b \bmod 3$ as $[b]_3$. For any $z \in \{-1, 0, 1\}$, define vector $\mathsf{enc}_3(z)$ in the following manner:

$$\mathsf{enc}_3(z) = \big([z+1]_3, [z]_3, [z-1]_3\big)^\top \in \{-1, 0, 1\}^3.$$

Namely, $\mathsf{enc}_3(-1) = (0, -1, 1)^\top$, $\mathsf{enc}_3(0) = (1, 0, -1)^\top$ and $\mathsf{enc}_3(1) = (-1, 1, 0)^\top$.

Let $e \in \{-1, 0, 1\}$, define a permutation π_e associated to e as follows. It transforms vector $\mathbf{v} = (v^{(-1)}, v^{(0)}, v^{(1)})^\top \in \mathbb{Z}^3$ into vector

$$\pi_e(\mathbf{v}) = (v^{([-e-1]_3)}, v^{([-e]_3)}, v^{([-e+1]_3)})^\top.$$

It is then verifiable that, for any $z, e \in \{-1, 0, 1\}$, the equivalence below holds.

$$\mathbf{v} = \mathsf{enc}_3(z) \iff \pi_e(\mathbf{v}) = \mathsf{enc}_3([z+e]_3). \tag{2}$$

In the context of Stern's protocol, the above equivalence allows us to prove knowledge of $z \in \{-1, 0, 1\}$, where z may have other constrains. Towards it, we

simply extend z to $\mathsf{enc}_3(z)$, sample a uniform $e \in \{-1,0,1\}$, and then show the verifier $\pi_e(\mathsf{enc}_3(z))$ is of the form $\mathsf{enc}_3([z+e]_3)$. Due to the equivalence in (2), the verifier should be convinced that z is in the set $\{-1,0,1\}$. Furthermore, the "one time pad" e fully hides the value of z. More importantly, the above technique is extendable so that we can employ the same e for other positions where z appears. An example of that is to prove that z is involved in a product $t \cdot z$, which we now recall.

Proving that $y = t \cdot z$. Let $b \in \{0,1\}$, denote the bit $1-b$ as $\overline{b}$ and the addition operation modulo 2 as $\oplus$.

For any $t \in \{0,1\}$ and $z \in \{-1,0,1\}$, let vector $\mathsf{ext}(t,z) \in \{-1,0,1\}^6$ be of the following form:

$$\mathsf{ext}(t,z) = \big(\ \overline{t} \cdot [z+1]_3,\ \ t \cdot [z+1]_3,\ \ \overline{t} \cdot [z]_3,\ \ t \cdot [z]_3,\ \ \overline{t} \cdot [z-1]_3,\ \ t \cdot [z-1]_3\ \big)^\top.$$

Let $b \in \{0,1\}$ and $e \in \{-1,0,1\}$, define the permutation $\psi_{b,e}(\cdot)$ associated to b, e as follows. It transforms vector

$$\mathbf{v} = \big(v^{(0,-1)}, v^{(1,-1)}, v^{(0,0)}, v^{(1,0)}, v^{(0,1)}, v^{(1,1)}\big)^\top \in \mathbb{Z}^6$$

into vector $\psi_{b,e}(\mathbf{v})$ of form

$$\psi_{b,e}(\mathbf{v}) = \big(v^{(b,[-e-1]_3)},\ v^{(\overline{b},[-e-1]_3)},\ v^{(b,[-e]_3)},\ v^{(\overline{b},[-e]_3)},\ v^{(b,[-e+1]_3)},\ v^{(\overline{b},[-e+1]_3)}\big)^\top.$$

It can be easily checked that for any $t, b \in \{0,1\}$ and any $z, e \in \{-1,0,1\}$, the following equivalence is satisfied.

$$\mathbf{v} = \mathsf{ext}(t,z) \iff \psi_{b,e}(\mathbf{v}) = \mathsf{ext}(\ t \oplus b,\ [z+e]_3\ ). \tag{3}$$

The same as in the case $z \in \{-1,0,1\}$, the above equivalence (3) allows us to prove knowledge of y, where y is a product of secret integers $t \in \{0,1\}$ and $z \in \{-1,0,1\}$.

Next, we recall the generalizations of the above two core techniques to prove knowledge of vector $\mathbf{z} \in \{-1,0,1\}^{\mathfrak{m}}$ as well as vector of the form (5).

Proving that $\mathbf{z} \in \{-1,0,1\}^{\mathfrak{m}}$. We first generalize the notion $[b]_3$ to $[\mathbf{b}]_3$ for any $\mathbf{b} \in \mathbb{Z}^{\mathfrak{m}}$, where $[\mathbf{b}]_3$ is the vector $\mathbf{b}'$ such that $\mathbf{b}' = \mathbf{b} \bmod 3$ coordinate-wise.

For $\mathbf{z} = (z_1, \ldots, z_{\mathfrak{m}})^\top \in \{-1,0,1\}^{\mathfrak{m}}$, define the following extension:

$$\mathsf{enc}(\mathbf{z}) = \big(\ \mathsf{enc}_3(z_1) \,\|\, \cdots \,\|\, \mathsf{enc}_3(z_{\mathfrak{m}})\ \big) \in \{-1,0,1\}^{3\mathfrak{m}}.$$

Let $\mathbf{e} = (e_1, \ldots, e_{\mathfrak{m}})^\top \in \{-1,0,1\}^{\mathfrak{m}}$, define the permutation $\mathit{\Pi}_{\mathbf{e}}$ associated to $\mathbf{e}$ as follows. It maps vector $\mathbf{v} = (\mathbf{v}_1 \| \ldots \| \mathbf{v}_{\mathfrak{m}}) \in \mathbb{Z}^{3\mathfrak{m}}$ consisting of $\mathfrak{m}$ blocks of size 3 to vector as follows:

$$\mathit{\Pi}_{\mathbf{e}}(\mathbf{v}) = \big(\pi_{e_1}(\mathbf{v}_1) \| \ldots \| \pi_{e_{\mathfrak{m}}}(\mathbf{v}_{\mathfrak{m}})\big).$$

Following (2), for any $\mathbf{z}, \mathbf{e} \in \{-1,0,1\}^{\mathfrak{m}}$, we obtain the following equivalence:

$$\mathbf{v} = \mathsf{enc}(\mathbf{z}) \iff \mathit{\Pi}_{\mathbf{e}}(\mathbf{v}) = \mathsf{enc}([\mathbf{z}+\mathbf{e}]_3). \tag{4}$$

Handling a "mixing" vector. We now deal with a "mixing" vector of the following form:

$$\mathbf{y} = \big(\mathbf{z} \parallel t_0 \cdot \mathbf{z} \parallel \ \ldots \ \parallel t_{c_d-1} \cdot \mathbf{z} \big), \tag{5}$$

where $\mathbf{z} \in \{-1,0,1\}^{\mathfrak{m}}$ and $t = (t_0, t_1, \ldots, t_{c_d-1})^\top \in \{0,1\}^{c_d}$ for $\mathfrak{m}, c_d \in \mathbb{Z}^+$.

First, we define the extension vector $\mathsf{mix}(\mathbf{t}, \mathbf{z}) \in \{-1,0,1\}^{3\mathfrak{m}+6\mathfrak{m}c_d}$ of vector $\mathbf{y}$ in the following manner:

$$\big(\mathsf{enc}(\mathbf{z}) \parallel \mathsf{ext}(t_0, z_1) \parallel \ldots \parallel \mathsf{ext}(t_0, z_{\mathfrak{m}}) \parallel \ldots \parallel \mathsf{ext}(t_{c_d-1}, z_1) \parallel \ldots \parallel \mathsf{ext}(t_{c_d-1}, z_{\mathfrak{m}}) \big).$$

Next, for $\mathbf{b} = (b_0, \cdots, b_{c_d-1})^\top \in \{0,1\}^{c_d}$ and $\mathbf{e} = (\mathbf{e}_1, \ldots, e_{\mathfrak{m}})^\top \in \{-1,0,1\}^{\mathfrak{m}}$, we define the permutation $\Psi_{\mathbf{b},\mathbf{e}}$ that works as follows. It maps vector $\mathbf{v} \in \mathbb{Z}^{3\mathfrak{m}+6\mathfrak{m}c_d}$ of form

$$\mathbf{v} = \big(\mathbf{v}_{-1} \parallel \mathbf{v}_{0,1} \parallel \ldots \parallel \mathbf{v}_{0,\mathfrak{m}} \parallel \ldots \parallel \mathbf{v}_{c_d-1,1} \parallel \ldots \parallel \mathbf{v}_{c_d-1,\mathfrak{m}}\big),$$

where block $\mathbf{v}_{-1}$ has length $3\mathfrak{m}$ and each block $\mathbf{v}_{i,j}$ has length 6, to vector $\Psi_{\mathbf{b},\mathbf{e}}(\mathbf{v})$ of form

$$\begin{aligned}\Psi_{\mathbf{b},\mathbf{e}}(\mathbf{v}) = \big(\Pi_{\mathbf{e}}(\mathbf{v}_{-1}) \| \ \psi_{b_0,e_1}(\mathbf{v}_{0,1}) \| \ldots \| \psi_{b_0,e_{\mathfrak{m}}}(\mathbf{v}_{0,\mathfrak{m}}) \| \ldots \| \\ \psi_{b_{c_d-1},e_1}(\mathbf{v}_{c_d-1,1}) \| \ldots \| \psi_{b_{c_d-1},e_{\mathfrak{m}}}(\mathbf{v}_{c_d-1,\mathfrak{m}})\big).\end{aligned}$$

Then, for all $\mathbf{t}, \mathbf{b} \in \{0,1\}^{c_d}$ and $\mathbf{z}, \mathbf{e} \in \{-1,0,1\}^{\mathfrak{m}}$, one can check the following equivalence holds:

$$\mathbf{v} = \mathsf{mix}(\mathbf{t}, \mathbf{z}) \iff \Psi_{\mathbf{b},\mathbf{e}}(\mathbf{v}) = \mathsf{mix}(\, \mathbf{t} \oplus \mathbf{b},\, [\mathbf{z} + \mathbf{e}]_3 \,). \tag{6}$$

2.6 Zero-Knowledge Protocol for the Ducas-Micciancio Signature

We now recall the statistical zero-knowledge argument of knowledge of a valid message-signature pair for the Ducas-Micciancio signature, as presented in [36]. Let $n, q, m, k, \overline{m}, \overline{m}_s, \ell, \beta, d, c_0, \ldots, c_d$ as specified in Section 2.3. The protocol is summarized below.

- The public input consists of

  $$\mathbf{A}, \mathbf{F}_0 \in R_q^{1\times\overline{m}}; \quad \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]} \in R_q^{1\times k}; \mathbf{F} \in R_q^{1\times\ell}; \quad \mathbf{F}_1 \in R_q^{1\times\overline{m}_s}; \quad u \in R_q.$$

- The secret input of the prover consists of message $\mathfrak{m} \in R_q^{m_s}$ and signature $(t, \mathbf{r}, \mathbf{v})$, where

  $$\begin{cases} t = (t_0, \ldots, t_{c_1-1}, \ldots, t_{c_{d-1}}, \ldots, t_{c_d-1})^\top \in \{0,1\}^{c_d}; \\ \mathbf{r} \in R^{\overline{m}}; \quad \mathbf{v} = (\mathbf{s} \| \mathbf{z}) \in R^{\overline{m}+k}; \quad \mathbf{s} \in R^{\overline{m}}; \quad \mathbf{z} \in R^k; \end{cases}$$

- The goal of the prover is to prove in ZK that $\|\mathbf{r}\|_\infty \le \beta$, $\|\mathbf{v}\|_\infty \le \beta$, and that the following equation

  $$\mathbf{A} \cdot \mathbf{s} + \mathbf{A}_{[0]} \cdot \mathbf{z} + \sum_{i=1}^{d} \mathbf{A}_{[i]} \cdot t_{[i]} \cdot \mathbf{z} = \mathbf{F} \cdot \mathbf{y} + u \tag{7}$$

holds for $\left\{t_{[i]} = \sum_{j=c_{i-1}}^{c_i-1} t_j \cdot X^j\right\}_{i=1}^{d}$ and

$$\mathbf{y} = \mathsf{rdec}\left(\mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \mathsf{rdec}(\mathfrak{m})\right) \in R^{\ell}. \tag{8}$$

The next step is to transform the secret input into a vector $\mathbf{w}$ that belongs to a specific set VALID and reduce the considered statements (7) and (8) into $\mathbf{M}\cdot\mathbf{w} = \mathbf{u} \bmod q$ for some public input $\mathbf{M}, \mathbf{u}$, in the form of the abstract protocol from Section 2.4. To realize this, we employ the following two steps.

Decomposing-Unifying. To begin with, we utilize the notations rot and τ from Section 2.1 and the decomposition techniques from Section 2.2.

Let $\mathbf{s}^\star = \tau(\mathsf{rdec}_\beta(\mathbf{s})) \in \{-1,0,1\}^{n\overline{m}\delta_\beta}$, $\mathbf{z}^\star = \tau(\mathsf{rdec}_\beta(\mathbf{z})) \in \{-1,0,1\}^{nk\delta_\beta}$ and $\mathbf{r}^\star = \tau(\mathsf{rdec}_\beta(\mathbf{r})) \in \{-1,0,1\}^{n\overline{m}\delta_\beta}$. Then, one can check that, equation (7) is equivalent to,

$$\begin{aligned}&[\mathsf{rot}(\mathbf{A}_{[0]}) \cdot \mathbf{H}_{k,\beta}] \cdot \mathbf{z}^\star + \sum_{i=1}^{d} \sum_{j=c_{i-1}}^{c_i-1} [\mathsf{rot}(\mathbf{A}_{[i]} \cdot X^j) \cdot \mathbf{H}_{k,\beta}] \cdot t_j \cdot \mathbf{z}^\star + \\ &[\mathsf{rot}(\mathbf{A}) \cdot \mathbf{H}_{\overline{m},\beta}] \cdot \mathbf{s}^\star - [\mathsf{rot}(\mathbf{F})] \cdot \tau(\mathbf{y}) = \tau(u) \bmod q,\end{aligned}$$

and equation (8) is equivalent to

$$[\mathsf{rot}(\mathbf{F}_0) \cdot \mathbf{H}_{\overline{m},\beta}] \cdot \mathbf{r}^\star + [\mathsf{rot}(\mathbf{F}_1)] \cdot \tau(\mathsf{rdec}(\mathfrak{m})) - [\mathbf{H}] \cdot \tau(\mathbf{y}) = \mathbf{0} \bmod q.$$

Rearrange the two derived equations using some basic algebra, we are able to obtain the following unifying equation:

$$\mathbf{M}_0 \cdot \mathbf{w}_0 = \mathbf{u} \bmod q,$$

where $\mathbf{u} = (\tau(u) \parallel \mathbf{0}) \in \mathbb{Z}_q^{2n}$ and $\mathbf{M}_0$ are built from public input, and $\mathbf{w}_0 = (\mathbf{w}_1 \parallel \mathbf{w}_2)$ is built from secret input with $\mathbf{w}_1 \in \{-1,0,1\}^{(k\delta_\beta + c_d k\delta_\beta)n}$ and $\mathbf{w}_2 \in \{-1,0,1\}^{2n\overline{m}\delta_\beta + n\ell + n\overline{m}_s}$ and

$$\begin{cases}\mathbf{w}_1 = (\mathbf{z}^\star \parallel t_0 \cdot \mathbf{z}^\star \parallel \ \ldots \ \parallel t_{c_d-1} \cdot \mathbf{z}^\star);\\ \mathbf{w}_2 = (\mathbf{s}^\star \parallel \mathbf{r}^\star \parallel \tau(\mathbf{y}) \parallel \tau(\mathsf{rdec}(\mathfrak{m}))).\end{cases}$$

Until now, we have transformed the secret input into a vector $\mathbf{w}_0$ whose coefficients are in the set $\{-1,0,1\}$ and reduced statements (7) and (8) into $\mathbf{M}_0 \cdot \mathbf{w}_0 = \mathbf{u} \bmod q$, where $\mathbf{M}_0, \mathbf{u}$ are public.

Extending-Permuting. Now the target is to transform the secret vector $\mathbf{w}_0$ to a vector $\mathbf{w}$ such that the conditions in (1) hold. Towards this goal, the extension and permutation techniques described in Section 2.5 is employed.

We first extend $\mathbf{w}_0 = (\mathbf{w}_1 \| \mathbf{w}_2)$ as follows.

$$\begin{aligned}&\mathbf{w}_1 \mapsto \mathbf{w}_1' = \mathsf{mix}\big(t, \mathbf{z}^\star\big) \in \{-1,0,1\}^{L_1};\\ &\mathbf{w}_2 \mapsto \mathbf{w}_2' = \mathsf{enc}(\mathbf{w}_2) \in \{-1,0,1\}^{L_2}.\end{aligned} \tag{9}$$

Then form a new vector $\mathbf{w} = (\mathbf{w}_1' \| \mathbf{w}_2') \in \{-1,0,1\}^L$, where $L = L_1 + L_2$ and

$$L_1 = (k\delta_\beta + 2c_d k\delta_\beta)3n;\ L_2 = 6n\overline{m}\delta_\beta + 3n\ell + 3n\overline{m}_s.$$

According to the extension, adding suitable zero-columns to $\mathbf{M}_0$ to obtain a new matrix $\mathbf{M} \in \mathbb{Z}_q^{2n \times L}$ such that $\mathbf{M} \cdot \mathbf{w} = \mathbf{M}_0 \cdot \mathbf{w}_0$.

We are ready to define the set VALID that consists of our transformed secret vector $\mathbf{w}$, the set $\mathcal{S}$, and the associated permutations $\{\Gamma_\eta : \eta \in \mathcal{S}\}$, such that the conditions in (1) are all satisfied.

Let VALID be the set of all vectors $\mathbf{v}' = (\mathbf{v}_1' \| \mathbf{v}_2') \in \{-1,0,1\}^L$ such that the following conditions hold:

- $\mathbf{v}_1' = \mathsf{mix}(t, \mathbf{z}^\star)$ for some vectors $t \in \{0,1\}^{c_d}$ and $\mathbf{z}^\star \in \{-1,0,1\}^{nk\delta_\beta}$.
- $\mathbf{v}_2' = \mathsf{enc}(\mathbf{w}_2)$ for vector $\mathbf{w}_2 \in \{-1,0,1\}^{L_2/3}$.

It is easy to see that $\mathbf{w}$ belongs to this special set VALID.

Now, define $\mathcal{S} = \{0,1\}^{c_d} \times \{-1,0,1\}^{nk\delta_\beta} \times \{-1,0,1\}^{L_2/3}$. For each element $\eta = (\mathbf{b}, \mathbf{e}, \mathbf{f}) \in \mathcal{S}$, define an associated permutation Γ_η as follows. It permutes vector $\mathbf{v}^\star = (\mathbf{v}_1^\star \| \mathbf{v}_2^\star) \in \mathbb{Z}^L$, where $\mathbf{v}_1^\star \in \mathbb{Z}^{L_1}$ and $\mathbf{v}_2^\star \in \mathbb{Z}^{L_2}$, into vector of the following form:

$$\Gamma_\eta(\mathbf{v}^\star) = \big(\Psi_{\mathbf{b},\mathbf{e}}(\mathbf{v}_1^\star) \parallel \Pi_{\mathbf{f}}(\mathbf{v}_2^\star) \big).$$

It then follows from the equivalences in (4) and (6) that VALID, $\mathcal{S}$, and Γ_η satisfy the conditions in (1). Therefore, we have obtained an instance of the abstract protocol from Section 2.4. Up to this point, running the protocol of Figure 1 results in the desired statistical ZKAoK protocol. The protocol has perfect completeness, soundness error 2/3, and communication cost $\mathcal{O}(L \cdot \log q)$, which is of order $\mathcal{O}(n \cdot \log^4 n) = \widetilde{\mathcal{O}}(\lambda)$.

2.7 Key-Oblivious Encryption

We next recall the definitions of key-oblivious encryption (KOE), as introduced in [25]. A KOE scheme consists of the following polynomial-time algorithms.

$\mathsf{Setup}(\lambda)$**:** On input the security parameter λ, it outputs public parameter pp. pp is implicit for all algorithms below if not explicitly mentioned.

$\mathsf{KeyGen}(\mathsf{pp})$**:** On input pp, it generates a key pair $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{KeyRand}(\mathsf{pk})$**:** On input the public key pk, it outputs a new public key pk' for the same secret key.

$\mathsf{Enc}(\mathsf{pk}, \mathfrak{m})$**:** On inputs pk and a message $\mathfrak{m}$, it outputs a ciphertext ct on this message.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$**:** On inputs sk and ct, it outputs the decrypted message $\mathfrak{m}'$.

Correctness. The above scheme must satisfy the following correctness requirement: For all λ, all $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$, all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, all $\mathsf{pk}' \leftarrow \mathsf{KeyRand}(\mathsf{pk})$, all $\mathfrak{m}$,

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}', \mathfrak{m})) = \mathfrak{m}.$$

SECURITY. The security requirements of a KOE scheme consist of *key randomizability* (KR), *plaintext indistinguishability under key randomization* (INDr), and *key privacy under key randomization* (KPr).

KEY RANDOMIZABILITY. KR requires that any adversary cannot determine how public keys are related to each other without possession of secret keys. Details are modelled in the experiment $\mathbf{Exp}^{\mathsf{KR}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ in Fig 2.

Define the advantage $\mathbf{Adv}^{\mathsf{KR}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against KR of the KOE scheme as $|2\Pr[\mathbf{Exp}^{\mathsf{KR}}_{\mathsf{KOE},\mathcal{A}}(\lambda) = 1] - 1|$. A KOE scheme is key randomizable if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

PLAINTEXT INDISTINGUISHABILITY UNDER KEY RANDOMIZATION. INDr requires that any adversary cannot distinguish ciphertext of one message from ciphertext of another one even though the adversary is allowed to choose the two messages and to randomize the public key. Details are modelled in the experiment $\mathbf{Exp}^{\mathsf{INDr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ in Fig 2.

Define the advantage $\mathbf{Adv}^{\mathsf{INDr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against INDr of the KOE scheme as $|2\Pr[\mathbf{Exp}^{\mathsf{INDr}}_{\mathsf{KOE},\mathcal{A}}(\lambda) = 1] - 1|$. A KOE scheme is plaintext indistinguishable under key randomization if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

$\underline{\mathbf{Exp}^{\mathsf{KR}}_{\mathsf{KOE},\mathcal{A}}(\lambda)}$
$b \leftarrow \{0,1\},\ \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda),\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}).$
$\mathsf{pk}_0 \leftarrow \mathsf{KeyRand}(\mathsf{pk}),\ (\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KeyGen}(\mathsf{pp}).$
$b' \leftarrow \mathcal{A}(\mathsf{pk},\mathsf{pk}_b).$
Return $(b' = b)$.
$\underline{\mathbf{Exp}^{\mathsf{INDr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)}$
$b \leftarrow \{0,1\},\ \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda),\ (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}).$
$(\mathsf{pk}', r, \mathtt{m}_0, \mathtt{m}_1, st) \leftarrow \mathcal{A}(\mathsf{pk}).$
If $\mathsf{pk}' \neq \mathsf{KeyRand}(\mathsf{pk}, r)$, then return $\perp$; else $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}', \mathtt{m}_b)$.
$b' \leftarrow \mathcal{A}(\mathsf{ct}, st).$
Return $(b' = b)$.
$\underline{\mathbf{Exp}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)}$
$b \leftarrow \{0,1\},\ \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KeyGen}(\mathsf{pp}),\ (\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KeyGen}(\mathsf{pp}).$
$(\mathtt{m}, \mathsf{pk}'_0, r_0, \mathsf{pk}'_1, r_1, st) \leftarrow \mathcal{A}(\mathsf{pk}_0,\mathsf{pk}_1).$
If $\exists\ c$ such that $\mathsf{pk}'_c \neq \mathsf{KeyRand}(\mathsf{pk}_c, r_c)$, then return $\perp$; else $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}'_b, \mathtt{m})$.
$b' \leftarrow \mathcal{A}(\mathsf{ct}, st).$
Return $(b' = b)$.

Fig. 2: Experiment to define security requirements of a KOE scheme.

KEY PRIVACY UNDER KEY RANDOMIZATION. KPr requires that any adversary cannot distinguish ciphertext of a message under one public key from ciphertext of the same message under another public key even though the adversary is allowed to choose the message and to randomize the two public keys. Details are modelled in the experiment $\mathbf{Exp}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ in Fig 2.

Define the advantage $\mathbf{Adv}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against INDr of the KOE scheme as $|2\Pr[\mathbf{Exp}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda) = 1] - 1|$. A KOE scheme is key private under key randomization if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

2.8 Accountable Tracing Signatures

We then recall the definition of accountable tracing signature (ATS), as introduced in [25]. An ATS scheme involves a group manager (GM) who also serves as the opening authority (OA), a set of users, who are potential group members. As a standard group signature scheme (e.g. [2,3]), GM is able to identify the signer of a given signature. However, if GM is able to do so, there is an additional *accounting* mechanism that later reveals which user he chose to trace (traceable user). Specifically, if a user suspects that he was traceable by group manager who had claimed non-traceability of this user, then the user can resort to this mechanism to check whether group manager is honest/accountable or not. An ATS scheme consists of the following polynomial-time algorithms.

$\mathsf{Setup}(\lambda)$**:** On input the security parameter λ, it outputs public parameter pp. pp is implicit for all algorithms below if not explicitly mentioned.

$\mathsf{GKeyGen}(\mathsf{pp})$**:** This algorithm is run by GM. On input pp, GM generates group public key gpk and group secret keys: issue key ik and opening key ok.

$\mathsf{UKeyGen}(\mathsf{pp})$**:** Given input pp, it outputs a user key pair $(\mathsf{upk}, \mathsf{usk})$.

$\mathsf{Enroll}(\mathsf{gpk}, \mathsf{ik}, \mathsf{upk}, \mathsf{tr})$**:** This algorithm is run by GM. Upon receiving a user public key upk from a user, GM determines the value of the bit $\mathsf{tr} \in \{0, 1\}$, indicating whether the user is traceable ($\mathsf{tr} = 1$) or not. He then produces a certificate cert for this user according to his choice of tr. GM then registers this user to the group and stores the registration information and the witness w^{escrw} to the bit tr, and sends cert to the user.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}, \mathsf{usk}, M)$**:** Given the inputs gpk, cert, usk and message M, this algorithm outputs a signature Σ on this message M.

$\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$**:** Given the inputs gpk and the message-signature pair (M, Σ), this algorithm outputs 1/0 indicating whether the signature is valid or not.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, M, \Sigma)$**:** Given the inputs gpk, ok and the pair (M, Σ), this algorithm returns a user public key upk' and a proof Π_{open} demonstrating that user upk' indeed generated the signature Σ. In case of $\mathsf{upk}' = \perp$, $\Pi_{\mathsf{open}} = \perp$.

$\mathsf{Judge}(\mathsf{gpk}, M, \Sigma, \mathsf{upk}', \Pi_{\mathsf{open}})$**:** Given all the inputs, this algorithm outputs 1/0 indicating whether it accepts the opening result or not.

$\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}, w^{\mathsf{escrw}}, \mathsf{tr})$**:** Given all the inputs, this algorithm returns 1 confirming the choice of tr and 0 otherwise.

Correctness. The above ATS scheme requires that: for any honestly generated signature, the Verify algorithm always outputs 1. Furthermore, if the user is traceable, then $\mathsf{Account}$ algorithm outputs 1 when $\mathsf{tr} = 1$, and the Open algorithm can identify the signer and generate a proof Π_{open} that will be accepted by the

Judge algorithm. On the other hand, if the user is non-traceable, then the Account algorithm outputs 1 when $\mathsf{tr} = 0$, and the Open algorithm outputs $\perp$.

Remark 1. There is a minor difference between the syntax we describe here and that presented by Kohlweiss and Miers [25]. Specifically, we omit the time epoch when the user joins the group, since we do not consider forward and backward tracing scenarios as in [25].

SECURITY. The security requirements of an ATS scheme consist of *anonymity under tracing* (AuT), *traceability* (Trace), and *non-frameability* (NF), *anonymity with accountability* (AwA) and *trace-obliviousness* (TO).

ANONYMITY UNDER TRACING. AuT is the standard anonymity requirement of group signatures (e.g. [2,3]). It guarantees that even when being traced, users are anonymous to the adversary who does not hold the opening key. Details are modelled in the experiment in Figure 3.

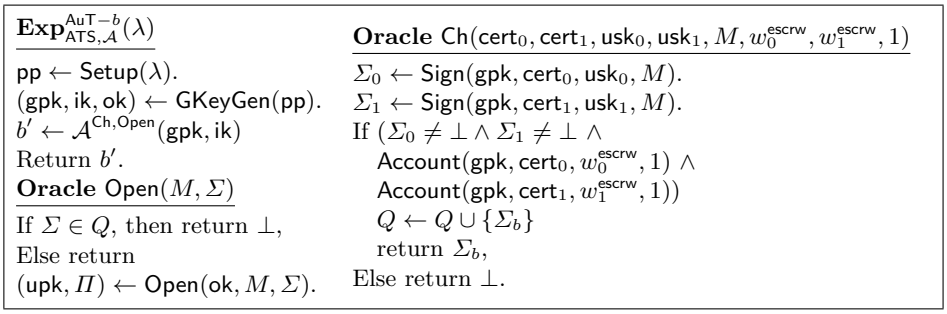

Fig. 3: Experiment to define anonymity under tracing

Define the advantage $\mathbf{Adv}^{\mathsf{AuT}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against anonymity under tracing of the ATS scheme as $|\Pr[\mathbf{Exp}^{\mathsf{AuT}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{AuT}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]|$. An ATS scheme is anonymous under tracing if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

TRACEABILITY. Traceability requires that every valid signature will trace to someone as long as the adversary does not hold both the certificate and user secret key of a user who is not traceable (non-traceable user). As pointed out by Kohlweiss and Miers [25], this is slightly different from the standard traceability game (e.g. [2,3]), where all users are being traced by GM. In an ATS scheme, when adversary queries certificate of a user of his choice, challenger will always generate a certificate according to $\mathsf{tr} = 1$. In other words, the user of the adversary's choice is a traceable user. This ensures that the adversary does not hold both certificate and user secret key for a non-traceable user. Details are modelled in the experiment in Figure 4.

Define the advantage $\mathbf{Adv}^{\mathsf{Trace}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against traceability of the ATS scheme as $\Pr[\mathbf{Exp}^{\mathsf{Trace}}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$. An ATS scheme is traceable if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

$\underline{\mathbf{Exp}^{\mathsf{Trace}}_{\mathsf{ATS},\mathcal{A}}(\lambda)}$

$\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$.
$(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}) \leftarrow \mathsf{GKeyGen}(\mathsf{pp})$.
$(M, \Sigma) \leftarrow \mathcal{A}^{\mathsf{UKG},\mathsf{Enroll},\mathsf{Sign},\mathsf{Open}}(\mathsf{gpk})$.
Return 0 if $(M, \Sigma) \in Q$ or
  $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma) = 0$.
Else $(\mathsf{upk}, \Pi) \leftarrow \mathsf{Open}(\mathsf{ok}, m, \Sigma)$.
  Return 1 if $\mathsf{upk} = \perp$ or
    $\mathsf{Judge}(\mathsf{gpk}, M, \Sigma, \mathsf{upk}, \Pi) = 0$.
  Else return 0.

$\underline{\textbf{Oracle } \mathsf{UKG}(\mathsf{pp})}$

$(\mathsf{upk}, \mathsf{usk}) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$.
$S[\mathsf{upk}] = \mathsf{usk}$.
Return upk.

$\underline{\textbf{Oracle } \mathsf{Enroll}(\mathsf{upk}, \mathsf{tr})}$

Let $\mathsf{tr}' = (\mathsf{upk} \notin \mathrm{dom}\ S) \in \{0, 1\}$.
$(\mathsf{cert}, w^{\mathsf{escrw}}) \leftarrow \mathsf{Enroll}(\mathsf{ik}, \mathsf{upk}, \mathsf{tr} \vee \mathsf{tr}')$.
Return cert.

$\underline{\textbf{Oracle } \mathsf{Sign}(\mathsf{cert}, M)}$

$\mathsf{usk} = S[\mathsf{cert.upk}]$.
If $(\mathsf{usk} = \perp)$, return $\perp$.
Else $\Sigma \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}, \mathsf{usk}, M)$.
  $Q = Q \cup \{(m, \Sigma)\}$.
  return Σ.

$\underline{\textbf{Oracle } \mathsf{Open}(M, \Sigma)}$

$(\mathsf{upk}, \Pi) \leftarrow \mathsf{Open}(\mathsf{ok}, M, \Sigma)$
Return (upk, Π).

Fig. 4: Experiment to define traceability.

Non-frameability. It requires that the adversary cannot sign messages on behalf of honest users, even though the adversary can corrupt GM and all other users. This ensures that signatures signed by a traceable user (traceable signatures) are non-repudiated. Details are modelled in the experiment in Figure 5.

$\underline{\mathbf{Exp}^{\mathsf{NF}}_{\mathsf{ATS},\mathcal{A}}(\lambda)}$

$\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$.
$(\mathsf{gpk}, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp})$.
If $\mathsf{gpk.pp} \neq \mathsf{pp}$, return $\perp$.
$(M, \Sigma, \mathsf{upk}, \Pi) \leftarrow \mathcal{A}^{\mathsf{UKG},\mathsf{Sign}}(\mathsf{st})$.
Return 1 if $((M, \Sigma) \notin Q \ \wedge$
  $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma) = 1 \ \wedge$
  $\mathsf{upk} \in \mathsf{dom}(S) \ \wedge$
  $\mathsf{Judge}(\mathsf{gpk}, M, \Sigma, \mathsf{upk}, \Pi) = 1)$.

$\underline{\textbf{Oracle } \mathsf{UKG}(\mathsf{pp})}$

$(\mathsf{upk}, \mathsf{usk}) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$,
$S[\mathsf{upk}] = \mathsf{usk}$.
Return upk.

$\underline{\textbf{Oracle } \mathsf{Sign}(\mathsf{cert}, M)}$

$\mathsf{usk} = S[\mathsf{cert.upk}]$.
If $(\mathsf{usk} = \perp)$ return $\perp$.
$\Sigma \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}, \mathsf{usk}, M)$.
$Q = Q \cup \{(M, \Sigma)\}$. Return Σ.

Fig. 5: Experiment to define non-frameability.

Define the advantage $\mathbf{Adv}^{\mathsf{NF}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against non-frameability of the ATS scheme as $\Pr[\mathbf{Exp}^{\mathsf{NF}}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$. An ATS scheme is non-frameable if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

Anonymity with accountability. AwA requires that a user is anonymous even from a corrupted group manager that has full control over the system as long as this user is non-traceable. In other words, the certificate is generated according to $\mathsf{tr} = 0$. Details are modelled in the experiment in Figure 6.

Define the advantage $\mathbf{Adv}^{\mathsf{AwA}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ of $\mathcal{A}$ against anonymity with accountability of the ATS scheme as $|\Pr[\mathbf{Exp}^{\mathsf{AwA}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{AwA}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]|$.

$\underline{\mathbf{Exp}^{\mathsf{AwA}-b}_{\mathsf{ATS},\mathcal{A}}(\lambda)}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$.
$(\mathsf{gpk}, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp})$.
If $\mathsf{gpk.pp} \neq \mathsf{pp}$, return $\perp$.
$b' \leftarrow \mathcal{A}^{\mathsf{Ch}}(\mathsf{st})$
Return b'.

$\underline{\textbf{Oracle } \mathsf{Ch}(\mathsf{cert}_0, \mathsf{cert}_1, \mathsf{usk}_0, \mathsf{usk}_1, M, w_0^{\mathsf{escrw}}, w_1^{\mathsf{escrw}}, 0)}$
$\Sigma_0 \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}_0, \mathsf{usk}_0, M)$.
$\Sigma_1 \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}_1, \mathsf{usk}_1, M)$.
If $(\Sigma_0 \neq \perp \wedge \Sigma_1 \neq \perp \wedge$
$\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}_0, w_0^{\mathsf{escrw}}, 0) \wedge$
$\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}_1, w_1^{\mathsf{escrw}}, 0))$,
return Σ_b.
Else return $\perp$.

Fig. 6: Experiment to define anonymity with accountability.

An ATS scheme is anonymous with accountability if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

TRACE-OBLIVIOUSNESS. Trace-obliviousness requires that each user cannot determine whether they are being traced or not. Details are modelled in the experiment in Figure 7.

$\underline{\mathbf{Exp}^{\mathsf{TO}-b}_{\mathsf{ATS},\mathcal{A}}(\lambda)}$
$\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$.
$(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}) \leftarrow \mathsf{GKeyGen}(\mathsf{pp})$.
$b' \leftarrow \mathcal{A}^{\mathsf{Ch},\mathsf{Enroll},\mathsf{Open}}(\mathsf{gpk})$
Return b'.

$\underline{\textbf{Oracle } \mathsf{Enroll}(\mathsf{upk}, \mathsf{tr})}$
$(\mathsf{cert}, w^{\mathsf{escrw}}) \leftarrow \mathsf{Enroll}(\mathsf{ik}, \mathsf{upk}, \mathsf{tr})$.
Return cert.
$\underline{\textbf{Oracle } \mathsf{Ch}(\mathsf{upk})}$
$(\mathsf{cert}, w^{\mathsf{escrw}}) \leftarrow \mathsf{Enroll}(\mathsf{ik}, \mathsf{upk}, b)$.
$U = U \cup \{\mathsf{upk}\}$, Return cert.
$\underline{\textbf{Oracle } \mathsf{Open}(M, \Sigma)}$
$(\mathsf{upk}, \Pi) \leftarrow \mathsf{Open}(\mathsf{ok}, M, \Sigma)$
If $\mathsf{upk} \in U$, then return $\perp$; Else return (upk, Π).

Fig. 7: Experiment to define trace-obliviousness.

Define the advantage $\mathbf{Adv}^{\mathsf{TO}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ of adversary $\mathcal{A}$ against trace-obliviousness of the ATS scheme as $|\Pr[\mathbf{Exp}^{\mathsf{TO}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{TO}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]|$. An ATS scheme is trace-oblivious if the advantage of any PPT adversary $\mathcal{A}$ is negligible.

3 Key-Oblivious Encryption from Lattices

In [25], Kohlweiss and Miers constructed a KOE scheme based on ElGamal cryptosystem [18]. To adapt their blueprint into the lattice setting, we would need a key-private homomorphic encryption scheme whose public keys and ciphertexts should have the same algebraic form (e.g., each of them is a pair of ring elements). We observe that, the LPR RLWE-based encryption scheme, under appropriate setting of parameters, does satisfy these conditions. We thus obtain an instantiation of KOE which will then serve as a building block for our ATS construction in Section 5.

3.1 Description of Our KOE Scheme

Our KOE scheme works as follows.

$\mathsf{Setup}(\lambda)$**:** Given the security parameter λ, let $n = \mathcal{O}(\lambda)$ be a power of 2 and $q = \widetilde{\mathcal{O}}(n^4)$. Also let $\ell = \lfloor \log \frac{q-1}{2} \rfloor + 1$. Define the rings $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = R/qR$. Let the integer bound B be of order $\widetilde{\mathcal{O}}(\sqrt{n})$ and χ be a B-bounded distribution over the ring R. This algorithm then outputs public parameter $\mathsf{pp} = \{n, q, \ell, R, R_q, B, \chi\}$.

$\mathsf{KeyGen}(\mathsf{pp})$**:** Given the input pp, this algorithm samples $s \hookleftarrow \chi$, $\mathbf{e} \hookleftarrow \chi^\ell$ and $\mathbf{a} \xleftarrow{\$} R_q^\ell$. Set $\mathsf{pk} = (\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \mathbf{a} \cdot s + \mathbf{e}) \in R_q^\ell \times R_q^\ell$ and $\mathsf{sk} = s$. It then returns $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{KeyRand}(\mathsf{pk})$**:** Given the public key $\mathsf{pk} = (\mathbf{a}, \mathbf{b})$, it samples $g \hookleftarrow \chi$, $\mathbf{e}_1 \hookleftarrow \chi^\ell$ and $\mathbf{e}_2 \hookleftarrow \chi^\ell$. Compute

$$(\mathbf{a}', \mathbf{b}') = (\mathbf{a} \cdot g + \mathbf{e}_1, \quad \mathbf{b} \cdot g + \mathbf{e}_2) \in R_q^\ell \times R_q^\ell.$$

This algorithm then outputs randomized public key as $\mathsf{pk}' = (\mathbf{a}', \mathbf{b}')$.

$\mathsf{Enc}(\mathsf{pk}', p)$**:** Given the public key $\mathsf{pk}' = (\mathbf{a}', \mathbf{b}')$ and a message $p \in R_q$, it samples $g' \in \chi$, $\mathbf{e}_1' \in \chi^\ell$ and $\mathbf{e}_2' \in \chi^\ell$. Compute

$$(\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{a}' \cdot g' + \mathbf{e}_1', \quad \mathbf{b}' \cdot g' + \mathbf{e}_2' + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p)) \in R_q^\ell \times R_q^\ell.$$

This algorithm returns ciphertext as $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$**:** Given $\mathsf{sk} = s$ and $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2)$, the algorithm proceeds as follows.

1. It computes
$$\mathbf{p}'' = \frac{\mathbf{c}_2 - \mathbf{c}_1 \cdot s}{\lfloor q/4 \rfloor}.$$
2. For each coefficient of $\mathbf{p}''$,
   - if it is closer to 0 than to -1 and 1, then round it to 0;
   - if it is closer to -1 than to 0 and 1, then round it to -1;
   - if it is closer to 1 than to 0 and -1, then round it to 1.
3. Denote the rounded $\mathbf{p}''$ as $\mathbf{p}' \in R_q^\ell$ with coefficients in $\{-1, 0, 1\}$.
4. Let $p' \in R_q$ such that $\tau(p') = \mathbf{H} \cdot \tau(\mathbf{p}')$. Here, $\mathbf{H} \in \mathbb{Z}_q^{n \times n\ell}$ is the decomposition matrix for elements of R_q (see Appendix 2.2).

3.2 Analysis of Our KOE Scheme

Correctness. Note that

$$\begin{aligned}\mathbf{c}_2 - \mathbf{c}_1 \cdot s &= \mathbf{b}' \cdot g' + \mathbf{e}_2' + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p) - (\mathbf{a}' \cdot g' + \mathbf{e}_1') \cdot s \\ &= \mathbf{e} \cdot g \cdot g' + \mathbf{e}_2 \cdot g' - \mathbf{e}_1 \cdot s \cdot g' + \mathbf{e}_2' - \mathbf{e}_1' \cdot s + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p)\end{aligned}$$

where $s, g, g', \mathbf{e}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1', \mathbf{e}_2'$ are B-bounded. Hence we have:

$$\|\mathbf{e} \cdot g \cdot g' + \mathbf{e}_2 \cdot g' - \mathbf{e}_1 \cdot s \cdot g' + \mathbf{e}_2' - \mathbf{e}_1' \cdot s\|_\infty \leq 3n^2 \cdot B^3 = \widetilde{\mathcal{O}}(n^{3.5}) \leq \lceil \frac{q}{10} \rceil = \widetilde{\mathcal{O}}(n^4).$$

With overwhelming probability, the rounding procedure described in the Dec algorithm recovers $\mathsf{rdec}(p)$ and hence outputs p. Therefore, our KOE scheme is correct.

SECURITY. The security of our KOE scheme is stated in the following theorem.

Theorem 2. *Under the* RLWE *assumption, the described key-oblivious encryption scheme satisfies: (i) key randomizability; (ii) plaintext indistinguishability under key randomization; and (iii) key privacy under key randomization.*

The proof of Theorem 2 is established by Lemma 2-4.

Lemma 2. *The key-oblivious encryption scheme described in Section 3.1 is key randomizable defined in Section 2.7 under* RLWE *assumption.*

Proof. Notice that the samples chosen according to $\mathcal{A}_{s,\chi}$ for some $s \hookleftarrow \chi$ are indistinguishable from random under the RLWE assumption. Therefore, the honestly generated public key $\mathsf{pk} = (\mathbf{a}, \mathbf{b}) \in R_q^\ell \times R_q^\ell$ is indistinguishable from truly random pair $\widetilde{\mathsf{pk}} = (\widetilde{\mathbf{a}}, \widetilde{\mathbf{b}}) \in R_q^\ell \times R_q^\ell$. Hence, we may replace pk with $\widetilde{\mathsf{pk}}$ and this modification is negligible to the adversary.

Let $\mathsf{pk}_0 = (\widetilde{\mathbf{a}} \cdot g + \mathbf{e}_1, \widetilde{\mathbf{b}} \cdot g + \mathbf{e}_2)$ and $\mathsf{pk}_1 = (\mathbf{a}', \mathbf{a}' \cdot s' + \mathbf{e}')$, where pk_1 is independent of $\widetilde{\mathsf{pk}}$. When $b = 0$, adversary is given $(\widetilde{\mathbf{a}}, \widetilde{\mathbf{b}}, \widetilde{\mathbf{a}} \cdot g + \mathbf{e}_1, \widetilde{\mathbf{b}} \cdot g + \mathbf{e}_2)$, which are 2ℓ samples chosen according to $\mathcal{A}_{g,\chi}$. Therefore, $(\widetilde{\mathsf{pk}}, \mathsf{pk}_0)$ is indistinguishable from 2ℓ samples chosen according to $U(R_q \times R_q)$. When $b = 1$, adversary is given $(\widetilde{\mathbf{a}}, \widetilde{\mathbf{b}}, \mathbf{a}', \mathbf{a}' \cdot s' + \mathbf{e}')$. Since pk_1 is independent of $\widetilde{\mathsf{pk}}$, so we can replace pk_1 with a truly random pair. Hence, $(\widetilde{\mathsf{pk}}, \mathsf{pk}_1)$ is also indistinguishable from 2ℓ samples chosen according to $U(R_q \times R_q)$. Therefore, the adversary cannot distinguish the case $b = 0$ from the case $b = 1$.

It then follows that the advantage of any PPT adversary in the experiment $\mathbf{Exp}^{\mathsf{KR}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ is negligible and hence our KOE scheme is key randomizable.

Lemma 3. *The key-oblivious encryption scheme described in Section 3.1 is plaintext indistinguishable under key randomization defined in Section 2.7 under* RLWE *assumption.*

Proof. Let $\mathcal{A}$ be any PPT adversary attacking the plaintext indistinguishability under key randomization with advantage ϵ, we will show $\epsilon = \mathrm{negl}(\lambda)$ assuming the hardness of the RLWE problem. Specifically, we construct a sequence of indistinguishable games G_0, G_1, G_2, G_3, G_4, such that, $\mathbf{Adv}_{\mathcal{A}}(G_0) = \epsilon$ and $\mathbf{Adv}_{\mathcal{A}}(G_4) = 0$.

Game G_0**:** This is the real experiment $\mathbf{Exp}^{\mathsf{INDr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$. The challenger generates a public key $\mathsf{pk} = (\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \mathbf{a} \cdot s + \mathbf{e})$ honestly, sends it to the adversary $\mathcal{A}$, receives back a randomized key pair $\mathsf{pk}' = (\mathbf{a} \cdot g + \mathbf{e}_1, \mathbf{b} \cdot g + \mathbf{e}_2)$, the randomness used to generate pk', and two messages $p_0, p_1 \in R_q$. The challenger first checks whether pk' is generated from the randomness or not. If not, the challenger returns $\perp$. Otherwise, he samples $b \xleftarrow{\$} \{0, 1\}$ and encrypts the

message p_b to ciphertext $(\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{a}' \cdot g' + \mathbf{e}'_1, \quad \mathbf{b}' \cdot g' + \mathbf{e}'_2 + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p_b))$ and sends $(\mathbf{c}_1, \mathbf{c}_2)$ to the adversary $\mathcal{A}$, who then outputs $b' \in \{0, 1\}$. This game outputs 1 if $b' = b$ or 0 otherwise. By assumption, $\mathcal{A}$ has advantage ϵ in this game.

Game G_1: In this game,we make a slight modification to the Game G_0: the public key pk is replaced with a truly random pair $\widetilde{\mathsf{pk}} = (\widetilde{\mathbf{a}}, \widetilde{\mathbf{b}})$. By the $\mathsf{RLWE}_{n,q,\ell,\chi}$ assumption, the adversary cannot distinguish $\mathsf{pk} = (\mathbf{a}, \mathbf{b})$ from uniform. It then follows that G_0 is indistinguishable from G_1. We additionally remark that pk' obtained from randomizing $\widetilde{\mathsf{pk}}$ is indistinguishable from random by the same assumption.

Game G_2: In this game, we modify G_1 as follows: instead of generating $(\mathbf{c}_1, \mathbf{c}_2)$ faithfully using the randomized public key pk', we generate ciphertext $(\mathbf{c}_1, \mathbf{c}_2)$ as $(\widetilde{\mathbf{a}'} \cdot g' + \mathbf{e}'_1, \quad \widetilde{\mathbf{b}'} \cdot g' + \mathbf{e}'_2 + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p_b))$, where $\widetilde{\mathsf{pk}'} = (\widetilde{\mathbf{a}'}, \widetilde{\mathbf{b}'})$ is uniformly chosen over $R_q^\ell \times R_q^\ell$. Since pk' obtained from randomizing $\widetilde{\mathsf{pk}}$ is indistinguishable from random, this modification is indistinguishable to adversary $\mathcal{A}$.

Game G_3: In this game, we generate $(\mathbf{c}_1, \mathbf{c}_2)$ as $(\mathbf{z}_1, \mathbf{z}_2 + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p_b))$, where $(\mathbf{z}_1, \mathbf{z}_2) \in R_q^\ell \times R_q^\ell$ are uniformly random. The assumed hardness of the $\mathsf{RLWE}_{n,q,\ell,\chi}$ problem implies that G_2 and G_3 are computationally indistinguishable.

Game G_4: In the game, we make a conceptual modification to G_3. Namely, we sample uniformly random $\mathbf{z}'_1 \in R_q^\ell$ and $\mathbf{z}'_2 \in R_q^\ell$ and let $(\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{z}'_1, \mathbf{z}'_2)$. It is clear that G_3 and G_4 are statistically indistinguishable. Moreover, since G_4 is no longer dependent on the challenger's bit b, the advantage of $\mathcal{A}$ in this game is 0.

It follows from the above construction that the advantage ϵ of the adversary $\mathcal{A}$ is negligible. This concludes the proof.

Lemma 4. *The key-oblivious encryption scheme described in Section 3.1 is key private under key randomization defined in Section 2.7 under* $\mathsf{RLWE}_{n,q,\chi}$ *assumption.*

Proof. The proof of Lemma 4 is similar to that of Lemma 3, we briefly describe it here. As in Lemma 3, we construct a sequence of indistinguishable games G_0, G_1, G_2, G_3, such that, $\mathbf{Adv}_{\mathcal{A}}(G_0) = \mathbf{Adv}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$ and $\mathbf{Adv}_{\mathcal{A}}(G_3) = 0$.

Game G_0 is the experiment $\mathbf{Exp}^{\mathsf{KPr}}_{\mathsf{KOE},\mathcal{A}}(\lambda)$, Game G_1 modifies Game G_0 by replacing public key pk_0 with truly random pair $\widetilde{\mathsf{pk}_0}$ while Game G_2 modifies Game G_1 by replacing public key pk_1 with another independent and random pair $\widetilde{\mathsf{pk}_1}$. By the hardness of the $\mathsf{RLWE}_{n,q,\ell,\chi}$ problem, these two modifications are indistinguishable to any PPT adversary. In Game G_3, we further modify Game G_2 by generating the ciphertext $(\mathbf{c}_1, \mathbf{c}_2)$ using $\widetilde{\mathsf{pk}'}$ chosen uniformly over $R_q^\ell \times R_q^\ell$ as in Lemma 3. By the same argument, this change is negligible to any PPT adversary. Furthermore, since G_3 is no longer dependent on the challenger's bit b, the advantage of adversary in this game is 0. This ends the brief description.

4 Handling Quadratically Hidden RLWE Relations

In Section 4.1, we extend the refined permuting technique recalled in Section 2.5 to prove that a secret integer y is multiplication of two secret integers $a \in \{-1, 0, 1\}$ and $g \in \{-1, 0, 1\}$. We then describe our zero-knowledge protocol for handling quadratic relations in the RLWE setting in Section 4.2. Specifically, we demonstrate how to prove in zero-knowledge that a give vector $\mathbf{c}$ is a correct RLWE evaluation, i.e., $\mathbf{c} = \mathbf{a} \cdot g + \mathbf{e}$, where the hidden vectors $\mathbf{a}, \mathbf{e}$ and element g may satisfy additional conditions. The protocol is developed based on Libert et al.'s work [29] on quadratic relations in the general lattice setting.

4.1 Our Extended Permuting Technique

Proving that $y = a \cdot g$. For any $a, g \in \{-1, 0, 1\}$, define vector $\mathsf{mult}_3(a, g) \in \{-1, 0, 1\}^9$ of the following form:

$$\mathsf{mult}_3(a,g) = \big([a+1]_3 \cdot [g+1]_3,\ [a]_3 \cdot [g+1]_3,\ [a-1]_3 \cdot [g+1]_3,\ [a+1]_3 \cdot [g]_3,$$
$$[a]_3 \cdot [g]_3,\ [a-1]_3 \cdot [g]_3\ , [a+1]_3 \cdot [g-1]_3,\ [a]_3 \cdot [g-1]_3,\ [a-1]_3 \cdot [g-1]_3\big)^\top.$$

Then for any $b, e \in \{-1, 0, 1\}$, we define the permutation $\phi_{b,e}(\cdot)$ that acts in the following way. It maps vector $\mathbf{v}$ of the following form

$$\mathbf{v} = \big(v^{(-1,-1)}, v^{(0,-1)}, v^{(1,-1)}, v^{(-1,0)}, v^{(0,0)}, v^{(1,0)}, v^{(-1,1)}, v^{(0,1)}, v^{(1,1)}\big)^\top \in \mathbb{Z}^9$$

into vector $\phi_{b,e}(\mathbf{v})$ of the following form

$$\begin{aligned}\phi_{b,e}(\mathbf{v}) = \big(&v^{([-b-1]_3,[-e-1]_3)}, v^{([-b]_3,[-e-1]_3)}, v^{([-b+1]_3,[-e-1]_3)},\\ &v^{([-b-1]_3,[-e]_3)}, v^{([-b]_3,[-e]_3)}, v^{([-b+1]_3,[-e]_3)},\\ &v^{([-b-1]_3,[-e+1]_3)}, v^{([-b]_3,[-e+1]_3)}, v^{([-b+1]_3,[-e+1]_3)}\big)^\top.\end{aligned}$$

Then for any $a, b, g, e \in \{-1, 0, 1\}$, one is able to check that the following equivalence is satisfied.

$$\mathbf{v} = \mathsf{mult}_3(a, g) \iff \phi_{b,e}(\mathbf{v}) = \mathsf{mult}_3([a+b]_3, [g+e]_3). \tag{10}$$

Note that the above equivalence in (10) is essential to prove knowledge of such secret integer y in the framework of Stern's protocol. We first extend y to vector $\mathbf{v} = \mathsf{mult}_3(a, g)$, sample uniform $b \in \{0, 1\}$ and $e \in \{-1, 0, 1\}$, and then demonstrate to the verifier $\phi_{b,e}(\mathbf{v}) = \mathsf{mult}_3(\,[a+b]_3,\, [g+e]_3\,)$. Due to the equivalence in (10), the verifier should be convinced of the well-formedness of y and no extra information is revealed to him. Furthermore, the technique is extendable so that we can use the same "one time pads" b and e at the places where a and g appear, respectively.

Now we generalize the above technique to prove knowledge of vector of the following expansion form. We aim to obtain equivalence similar to (10), which is useful in Stern's framework.

Handling an expansion vector. We now tackle an expansion vector $\mathbf{y} = \mathsf{expd}(\mathbf{a}, \mathbf{g})$ of the form $\mathbf{y} = (\mathbf{y}_0 \| \dots \| \mathbf{y}_{n-1}) \in \{-1, 0, 1\}^{n^2 \ell \delta_B}$, where $\mathbf{y}_i$ is of the following form

$$\mathbf{y}_i = (a_1 \cdot g_{i,1},\ \dots,\ a_1 \cdot g_{i,\delta_B},\ \dots,\ a_{n\ell} \cdot g_{i,1}\ ,\dots,\ a_{n\ell} \cdot g_{i,\delta_B}),$$

$\mathbf{g} \in \{-1, 0, 1\}^{n\delta_B}$ is of the form

$$\mathbf{g} = (g_{0,1}, g_{0,2}, \dots, g_{0,\delta_B}, \dots, g_{n-1,1}, g_{n-1,2}, \dots, g_{n-1,\delta_B})^\top,$$

and $\mathbf{a} = (a_1, \dots, a_{n\ell})^\top \in \{-1, 0, 1\}^{n\ell}$ for some positive integers n, ℓ, δ_B.

Denote $\mathbf{y} = (a_i \cdot g_{j,k})_{i \in [n\ell], j \in [0,n-1], k \in [\delta_B]}$, we then define an extension of the expansion vector $\mathbf{y}$ as $\mathsf{mult}(\mathbf{a}, \mathbf{g}) = (\mathsf{mult}_3(a_i, g_{j,k}))_{i \in [n\ell], j \in [0,n-1], k \in [\delta_B]} \in \{-1, 0, 1\}^{9n^2 \ell \delta_B}$.

For $\mathbf{e} = (e_{0,1}, e_{0,2}, \dots, e_{0,\delta_B}, \dots, e_{n-1,1}, e_{n-1,2}, \dots, e_{n-1,\delta_B})^\top \in \{-1, 0, 1\}^{n\delta_B}$ and $\mathbf{b} = (b_1, \dots, b_{n\ell})^\top \in \{-1, 0, 1\}^{n\ell}$, we define the permutation $\Phi_{\mathbf{b},\mathbf{e}}(\cdot)$ that behaves as follows. It maps vector $\mathbf{v} \in \mathbb{Z}^{9n^2 \ell \delta_B}$ of the following form:

$$\begin{aligned}
&\big(\mathbf{v}_{1,0,1} \| \cdots \| \mathbf{v}_{1,0,\delta_B} \| \cdots \| \mathbf{v}_{n\ell,0,1} \| \cdots \| \mathbf{v}_{n\ell,0,\delta_B} \| \\
&\ \mathbf{v}_{1,1,1} \| \cdots \| \mathbf{v}_{1,1,\delta_B} \| \cdots \| \mathbf{v}_{n\ell,1,1} \| \cdots \| \mathbf{v}_{n\ell,1,\delta_B} \| \\
&\ \cdots\cdot \\
&\ \mathbf{v}_{1,n-1,1} \| \cdots \| \mathbf{v}_{1,n-1,\delta_B} \| \cdots \| \mathbf{v}_{n\ell,n-1,1} \| \cdots \| \mathbf{v}_{n\ell,n-1,\delta_B}\big)
\end{aligned}$$

which consists of blocks of size 9, to vector $\Phi_{\mathbf{b},\mathbf{e}}(\mathbf{v})$ of the following form:

$$\begin{aligned}
&\big(\phi_{b_1,e_{0,1}}(\mathbf{v}_{1,0,1}) \| \cdots \| \phi_{b_1,e_{0,\delta_B}}(\mathbf{v}_{1,0,\delta_B}) \| \cdots \| \\
&\ \phi_{b_{n\ell},e_{0,1}}(\mathbf{v}_{n\ell,0,1}) \| \cdots \| \phi_{b_{n\ell},e_{0,\delta_B}}(\mathbf{v}_{n\ell,0,\delta_B}) \| \\
&\ \phi_{b_1,e_{1,1}}(\mathbf{v}_{1,1,1}) \| \cdots \| \phi_{b_1,e_{1,\delta_B}}(\mathbf{v}_{1,1,\delta_B}) \| \cdots \| \\
&\ \phi_{b_{n\ell},e_{1,1}}(\mathbf{v}_{n\ell,1,1}) \| \cdots \| \phi_{b_{n\ell},e_{1,\delta_B}}(\mathbf{v}_{n\ell,1,\delta_B}) \| \\
&\ \cdots\cdot \\
&\ \phi_{b_1,e_{n-1,1}}(\mathbf{v}_{1,n-1,1}) \| \cdots \| \phi_{b_1,e_{n-1,\delta_B}}(\mathbf{v}_{1,n-1,\delta_B}) \| \cdots \| \\
&\ \phi_{b_{n\ell},e_{n-1,1}}(\mathbf{v}_{n\ell,n-1,1}) \| \cdots \| \phi_{b_{n\ell},e_{n-1,\delta_B}}(\mathbf{v}_{n\ell,n-1,\delta_B})\big)
\end{aligned}$$

For any $\mathbf{a}, \mathbf{b} \in \{-1, 0, 1\}^{n\ell}$ and any $\mathbf{g}, \mathbf{e} \in \{-1, 0, 1\}^{n\delta_B}$, it then follows from (10) that the following equivalence holds.

$$\mathbf{v} = \mathsf{mult}(\mathbf{a}, \mathbf{g}) \iff \Phi_{\mathbf{b},\mathbf{e}}(\mathbf{v}) = \mathsf{mult}([\mathbf{a} + \mathbf{b}]_3, [\mathbf{g} + \mathbf{e}]_3). \tag{11}$$

4.2 Proving the RLWE Relation with Hidden Vector

We are going to describe our statistical ZKAoK protocol for the RLWE relation with hidden vector. Let q, ℓ, B be some integers and R, R_q be two rings, which are specified as in Section 3.1. Our goal is to design a ZK argument system that

allows a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ on input $\mathbf{c} \in R_q^\ell$ that $\mathcal{P}$ knows secrets $\mathbf{a} \in R_q^\ell$, $g \in R_q$ and $\mathbf{e} \in R_q^\ell$ such that g and $\mathbf{e}$ are B-bounded and

$$\mathbf{c} = \mathbf{a} \cdot g + \mathbf{e}. \tag{12}$$

Furthermore, this protocol should be extendable such that we are able to prove that the secrets $\mathbf{a}, g, \mathbf{e}$ satisfy other relations.

As in Section 2.6, we aim to obtain an instance of the abstract protocol from Section 2.4.

Decomposing-Unifying. To start with, we also employ the notations rot and τ from Section 2.1 and the decomposition techniques from Section 2.2 to transform equation (12) into $\mathbf{M}_0 \cdot \mathbf{w}_0 = \mathbf{u} \bmod q$, where $\mathbf{M}_0, \mathbf{u}$ are built from public input, and vector $\mathbf{w}_0$ is built from secret input and coefficients of which are in the set $\{-1, 0, 1\}$.

Let $\mathbf{a} = (a_1, a_2, \cdots, a_\ell)^\top$, $\tau(g) = (g_0, \cdots, g_{n-1})^\top$, $\mathbf{a}_i^\star = \tau(\mathsf{rdec}(a_i)) \in \{-1, 0, 1\}^{n\ell}$ $\forall\ i \in [\ell]$, $\mathbf{g}^\star = \tau(\mathsf{rdec}_B(g)) \in \{-1, 0, 1\}^{n\delta_B}$. Let $\mathbf{a}_i^\star = (a_{i,1}, a_{i,2}, \cdots, a_{i,n\ell})^\top$ $\forall\ i \in [\ell]$, $\mathbf{g}^\star = (g_{0,1}, \cdots g_{0,\delta_B}, \cdots, g_{n-1,1}, \cdots, g_{n-1,\delta_B})^\top$. We then have the following:

$$\begin{aligned}
\tau(a_i \cdot g) &= \mathsf{rot}(a_i) \cdot \tau(g) = [\tau(a_i) | \tau(a_i \cdot X) | \ldots | \tau(a_i \cdot X^{n-1})] \cdot \tau(g) \\
&= \sum_{j=0}^{n-1} \tau(a_i \cdot X^j) \cdot g_j = \sum_{j=0}^{n-1} \mathsf{rot}(X^j) \cdot \tau(a_i) \cdot g_j = \sum_{j=0}^{n-1} \mathsf{rot}(X^j) \cdot \mathbf{H} \cdot \mathbf{a}_i^\star \cdot g_j \\
&= \sum_{j=0}^{n-1} \mathsf{rot}(X^j) \cdot \mathbf{H} \cdot (a_{i,1} \cdot g_j, \ldots, a_{i,n\ell} \cdot g_j)^\top \bmod q
\end{aligned}$$

Observe that, for each $k \in [n\ell]$, we have

$$\begin{aligned}
a_{i,k} \cdot g_j &= a_{i,k} \cdot (B_1, \ldots, B_{\delta_B}) \cdot (g_{j,1}, \ldots, g_{j,\delta_B})^\top \\
&= (B_1, \ldots, B_{\delta_B}) \cdot (a_{i,k} \cdot g_{j,1}, \ldots, a_{i,k} \cdot g_{j,\delta_B})^\top
\end{aligned}$$

Denote $\mathbf{y}_{i,j} \in \{-1, 0, 1\}^{n\ell\delta_B}$ of the following form:

$$\mathbf{y}_{i,j} = (a_{i,1} \cdot g_{j,1}, \ldots, a_{i,1} \cdot g_{j,\delta_B}, \ldots, a_{i,n\ell} \cdot g_{j,1}, \ldots, a_{i,n\ell} \cdot g_{j,\delta_B})^\top,$$

we then obtain

$$(a_{i,1} \cdot g_j, \ldots, a_{i,n\ell} \cdot g_j)^\top = \mathbf{H}_{\ell,B} \cdot \mathbf{y}_{i,j} \mod q.$$

Define $\mathbf{Q}_0 \in \mathbb{Z}_q^{n \times n^2\ell\delta_B}$ of the following form:

$$\mathbf{Q}_0 = [\mathsf{rot}(X^0) \cdot \mathbf{H} \cdot \mathbf{H}_{\ell,B} | \cdots | \mathsf{rot}(X^{n-1}) \cdot \mathbf{H} \cdot \mathbf{H}_{\ell,B}].$$

Let $\mathbf{y}_i = (\mathbf{y}_{i,0} \| \cdots \| \mathbf{y}_{i,n-1}) = \mathsf{expd}(\mathbf{a}_i^\star, \mathbf{g}^\star) \in \{-1, 0, 1\}^{n^2\ell\delta_B}$, we then obtain:

$$\tau(a_i \cdot g) = \mathbf{Q}_0 \cdot \mathbf{y}_i \mod q.$$

Let $\mathbf{e}^\star = \tau(\mathsf{rdec}_B(\mathbf{e})) \in \{-1,0,1\}^{n\ell\delta_B}$, $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_0 & & & \\ & \mathbf{Q}_0 & & \\ & & \ddots & \\ & & & \mathbf{Q}_0 \end{pmatrix} \in \mathbb{Z}_q^{n\ell \times n^2\ell^2\delta_B}$.

Now equation (12) is equivalent to

$$\begin{aligned}\tau(\mathbf{c}) &= (\tau(a_1 \cdot g), \ldots, \tau(a_\ell \cdot g))^\top + \tau(\mathbf{e}) \\ &= \mathbf{Q} \cdot (\mathbf{y}_1 \| \cdots \| \mathbf{y}_\ell) + \mathbf{H}_{\ell,B} \cdot \mathbf{e}^\star \bmod q\end{aligned}$$

Rearrange the above equivalent form using some basic algebra, we are able to obtain an unifying equation of the following form:

$$\mathbf{M}_0 \cdot \mathbf{w}_0 = \mathbf{u} \bmod q,$$

where $\mathbf{M}_0$ is built from the public matrices $\mathbf{Q}$ and $\mathbf{H}_{\ell,B}$, $\mathbf{u}$ is the vector $\tau(\mathbf{c})$, while $\mathbf{w}_0 = (\mathbf{y}_1 \| \cdots \| \mathbf{y}_\ell \| \mathbf{e}^\star) \in \{-1,0,1\}^{n^2\ell^2\delta_B + n\ell\delta_B}$.

Extending-Permuting. In this second step, we aim to transform the secret $\mathbf{w}_0$ to a vector $\mathbf{w}$ such that it satisfies the requirements specified by the abstract protocol from section 2.4. In the process, the techniques introduced in Section 2.5 and 4.1 are utilized.

We first extend $\mathbf{w}_0 = (\mathbf{y}_1 \| \cdots \| \mathbf{y}_\ell \| \mathbf{e}^\star)$ as follows.

$$\begin{aligned}&\mathbf{y}_i \mapsto \mathbf{y}'_i = \mathsf{mult}\big(\mathbf{a}_i^\star, \mathbf{g}^\star\big) \in \{-1,0,1\}^{9n^2\ell\delta_B},\ i \in [\ell]; \\ &\mathbf{e}^\star \mapsto \mathbf{e}'^\star = \mathsf{enc}(\mathbf{e}^\star) \in \{-1,0,1\}^{L_2}.\end{aligned}$$

Notice that for each $i \in [\ell]$, we have $\mathbf{y}_i = \mathsf{expd}(\mathbf{a}_i^\star, \mathbf{g}^\star)$. We then form vector $\mathbf{w} = (\mathbf{y}'_1 \| \cdots \| \mathbf{y}'_\ell \| \mathbf{e}'^\star) \in \{-1,0,1\}^L$, where

$$L = L_1 + L_2; \quad L_1 = 9n^2\ell^2\delta_B; \quad L_2 = 3n\ell\delta_B.$$

According to the extension, we insert appropriate zero-columns to matrix $\mathbf{M}_0$, obtaining a new matrix $\mathbf{M} \in \mathbb{Z}_q^{n\ell \times L}$ such that the equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{M}_0 \cdot \mathbf{w}_0$ holds.

We now define the set VALID that includes our secret vector $\mathbf{w}$, the set $\mathcal{S}$, and the associated permutations $\{\Gamma_\eta : \eta \in \mathcal{S}\}$, such that the conditions in (1) are satisfied.
Let VALID be the set of all vectors $\mathbf{v}' = (\mathbf{v}'_1 \| \cdots \| \mathbf{v}'_\ell \| \mathbf{v}'_{\ell+1}) \in \{-1,0,1\}^L$ such that the following conditions hold:

- There exist $\mathbf{a}_i^\star \in \{-1,0,1\}^{n\ell}$ for each $i \in [\ell]$ and $\mathbf{g}^\star \in \{-1,0,1\}^{n\delta_B}$ such that $\mathbf{v}'_i = \mathsf{mult}(\mathbf{a}_i^\star, \mathbf{g}^\star)$.
- There exists $\mathbf{e}^\star \in \{-1,0,1\}^{n\ell\delta_B}$ such that $\mathbf{v}'_{\ell+1} = \mathsf{enc}(\mathbf{e}^\star)$.

It is easy to see that the obtained vector $\mathbf{w}$ belongs to the set VALID.

Now let $\mathcal{S} = (\{-1,0,1\}^{n\ell})^\ell \times \{-1,0,1\}^{n\delta_B} \times \{-1,0,1\}^{n\ell\delta_B}$, and associate every element $\eta = (\mathbf{b}_1, \ldots, \mathbf{b}_\ell, \mathbf{f}_1, \mathbf{f}_2) \in \mathcal{S}$ with permutation Γ_η that behaves as

follows. For a vector of the form $\mathbf{v} = (\mathbf{v}_1 \| \cdots \| \mathbf{v}_\ell \| \mathbf{v}_{\ell+1}) \in \mathbb{Z}^L$, where $\mathbf{v}_i \in \mathbb{Z}^{9n^2\ell\delta_B}$ for each $i \in [\ell]$ and $\mathbf{v}_{\ell+1} \in \mathbb{Z}^{L_2}$, it transforms $\mathbf{v}$ into vector

$$\Gamma_\eta(\mathbf{v}) = \big(\Phi_{\mathbf{b}_1,\mathbf{f}_1}(\mathbf{v}_1) \,\|\, \cdots \,\|\, \Phi_{\mathbf{b}_\ell,\mathbf{f}_1}(\mathbf{v}_\ell) \,\|\, \Pi_{\mathbf{f}_2}(\mathbf{v}_{\ell+1}) \big).$$

It then follows from the equivalences in (4) and (11) that VALID, $\mathcal{S}$, and Γ_η fulfill the requirements specified in (1). Therefore, we have transformed the considered statement to a case of the abstract protocol from Section 2.4. To obtain the desired statistical ZKAoK protocol, it suffices for the prover and verifier to run the interactive protocol described in Figure 1. The protocol has perfect completeness, soundness error 2/3 and communication cost $\mathcal{O}(L \cdot \log q)$, which is of order $\mathcal{O}(n^2 \cdot \log^4 n) = \widetilde{\mathcal{O}}(\lambda^2)$.

5 Accountable Tracing Signatures from Lattices

In this section, we construct our ATS scheme based on: (i) The Ducas-Micciancio signature scheme (as recalled in Section 2.3); (ii) The KOE scheme described in Section 3; and (iii) Stern-like zero-knowledge argument system that underlies our ATS construction, which is obtained by smoothly combining previous techniques as recalled in Section 2.6 and ours as described in Section 4.2.

5.1 The Zero-Knowledge Argument System Underlying the ATS Scheme

Before describing our accountable tracing signature scheme in Section 5.2, let us first present the statistical ZKAoK that will be invoked by the signer when generating group signatures. Let $n, q, k, \ell, m, \overline{m}, \overline{m}_s, d, c_0, \cdots, c_d, \beta, B$ be parameters as specified in Section 5.2. The protocol is summarized as follows.

– The public input consists of

$$\mathbf{A}, \mathbf{F}_0 \in R_q^{1\times\overline{m}}; \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]} \in R_q^{1\times k}; \mathbf{F} \in R_q^{1\times\ell};$$
$$\mathbf{F}_1 \in R_q^{1\times\overline{m}_s}; u \in R_q; \mathbf{B} \in R_q^m; \mathbf{c}_{1,1}, \mathbf{c}_{1,2} \in R_q^\ell, \mathbf{c}_{2,1}, \mathbf{c}_{2,2} \in R_q^\ell.$$

– The secret input of the prover consists of message $\mathfrak{m} = (p \| \mathbf{a}'_1 \| \mathbf{b}'_1 \| \mathbf{a}'_2 \| \mathbf{b}'_2)$ and the corresponding Ducas-Micciancio signature $(t, \mathbf{r}, \mathbf{v})$, a user secret key $\mathbf{x}$ that corresponds to the public key p, and encryption randomness $g'_1, g'_2, \mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}, \mathbf{e}'_{2,1}, \mathbf{e}'_{2,2}$, where

$$\begin{cases} p \in R_q; \;\; \mathbf{a}'_1 \in R_q^\ell; \;\; \mathbf{b}'_1 \in R_q^\ell; \;\; \mathbf{a}'_2 \in R_q^\ell; \;\; \mathbf{b}'_2 \in R_q^\ell; \\ t = (t_0, \ldots, t_{c_1-1}, \ldots, t_{c_{d-1}}, \ldots, t_{c_d-1})^\top \in \{0,1\}^{c_d}; \\ \mathbf{r} \in R^{\overline{m}}; \;\; \mathbf{v} = (\mathbf{s} \| \mathbf{z}) \in R^{\overline{m}+k}; \;\; \mathbf{s} \in R^{\overline{m}}; \;\; \mathbf{z} \in R^k; \\ \mathbf{x} \in R^m; \;\; g'_1, g'_2 \in R; \;\; \mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}, \mathbf{e}'_{2,1}, \mathbf{e}'_{2,2} \in R^\ell. \end{cases}$$

– The goal of the prover is to prove in ZK that $\|\mathbf{r}\|_\infty \leq \beta$, $\|\mathbf{v}\|_\infty \leq \beta$, $\|\mathbf{x}\|_\infty \leq 1$, $\|g'_i\|_\infty \leq B$, $\|\mathbf{e}_{i,1}\|_\infty \leq B$, $\|\mathbf{e}_{i,2}\|_\infty \leq B$ and that the following conditions hold:

$$\mathbf{A}_t \cdot \mathbf{v} = \mathbf{F} \cdot \mathsf{rdec}\left(\mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \mathsf{rdec}(\mathfrak{m})\right) + u,$$

$$\mathbf{B} \cdot \mathbf{x} = p,$$

$$\text{for } i \in \{1,2\},\ \mathbf{c}_{i,1} = \mathbf{a}'_i \cdot g'_i + \mathbf{e}'_{i,1},\quad \mathbf{c}_{i,2} = \mathbf{b}'_i \cdot g'_i + \mathbf{e}'_{i,2} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p). \tag{13}$$

Since we already established the transformations for the Ducas-Micciancio signature in Section 2.6, we now focus on the transformations for other relations.

Let $\mathbf{a}'_i = (a'_{i,1}, \ldots, a'_{i,\ell})^\top$, $\mathbf{b}'_i = (b'_{i,1}, \ldots, b'_{i,\ell})\top$ for each $i \in \{1,2\}$. First, we employ the decomposition techniques in Section 2.2 to the following secrets.

– Let $\mathbf{x}^\star = \tau(\mathbf{x}) \in \{-1,0,1\}^{nm}$.
– For each $i \in \{1,2\}$, each $j \in [\ell]$, compute $\mathbf{a}^\star_{i,j} = \tau(\mathsf{rdec}(a'_{i,j})) \in \{-1,0,1\}^{n\ell}$, $\mathbf{b}^\star_{i,j} = \tau(\mathsf{rdec}(b'_{i,j})) \in \{-1,0,1\}^{n\ell}$.
– For $i \in \{1,2\}$, compute $\mathbf{g}^\star_i = \tau(\mathsf{rdec}_B(g'_i)) \in \{-1,0,1\}^{n\delta_B}$.
– For $i \in \{1,2\}$, compute $\mathbf{e}^\star_{i,1} = \tau(\mathsf{rdec}_B(\mathbf{e}'_{i,1})) \in \{-1,0,1\}^{n\ell\delta_B}$ and $\mathbf{e}^\star_{i,2} = \tau(\mathsf{rdec}_B(\mathbf{e}'_{i,2})) \in \{-1,0,1\}^{n\ell\delta_B}$.

Then the equation $\mathbf{B} \cdot \mathbf{x} = p$ over R_q is equivalent to

$$[\mathsf{rot}(\mathbf{B})] \cdot \mathbf{x}^\star - [\mathbf{H}] \cdot \tau(\mathsf{rdec}(p)) = \mathbf{0}^n \bmod q. \tag{14}$$

For each $i \in \{1,2\}$, each $j \in [\ell]$, let

$$\begin{cases} \mathbf{y}_{i,j} = \mathsf{expd}\left(\mathbf{a}^\star_{i,j},\ \mathbf{g}^\star_i\right) \in \{-1,0,1\}^{n^2\ell\delta_B}, \\ \mathbf{z}_{i,j} = \mathsf{expd}\left(\mathbf{b}^\star_{i,j},\ \mathbf{g}^\star_i\right) \in \{-1,0,1\}^{n^2\ell\delta_B}. \end{cases} \tag{15}$$

From Section 4.2, we know that equations in (13) can be written as, for $i \in \{1,2\}$,

$$\begin{cases} \tau(\mathbf{c}_{i,1}) = [\mathbf{Q}] \cdot (\mathbf{y}_{i,1} \| \cdots \| \mathbf{y}_{i,\ell}) + [\mathbf{H}_{\ell,B}] \cdot \mathbf{e}^\star_{i,1}; \\ \tau(\mathbf{c}_{i,2}) = [\mathbf{Q}] \cdot (\mathbf{z}_{i,1} \| \cdots \| \mathbf{z}_{i,\ell}) + [\mathbf{H}_{\ell,B}] \cdot \mathbf{e}^\star_{i,2} + \lfloor q/4 \rfloor \cdot \tau(\mathsf{rdec}(p)). \end{cases} \tag{16}$$

Following the procedure in Section 2.6, we form secret vectors $\mathbf{w}_1 \in \{-1,0,1\}^{(k\delta_\beta + c_d k\delta_\beta)n}$, $\mathbf{w}_2 \in \{-1,0,1\}^{2n\overline{m}\delta_\beta + n\ell + n\overline{m}_s}$ of the form:

$$\begin{cases} \mathbf{w}_1 = (\mathbf{z}^\star \parallel t_0 \cdot \mathbf{z}^\star \parallel \ \ldots \ \parallel t_{c_d - 1} \cdot \mathbf{z}^\star); \\ \mathbf{w}_2 = (\mathbf{s}^\star \parallel \mathbf{r}^\star \parallel \tau(\mathbf{y}) \parallel \tau(\mathsf{rdec}(\mathfrak{m}))), \end{cases}$$

where $\tau(\mathsf{rdec}(\mathfrak{m}))$

$$= (\tau(\mathsf{rdec}(p)) \| \tau(\mathsf{rdec}(\mathbf{a}'_1)) \| \tau(\mathsf{rdec}(\mathbf{b}'_1)) \| \tau(\mathsf{rdec}(\mathbf{a}'_2)) \| \tau(\mathsf{rdec}(\mathbf{b}'_1)))$$
$$= (\tau(\mathsf{rdec}(p)) \| \mathbf{a}^\star_{1,1} \| \cdots \| \mathbf{a}^\star_{1,\ell} \| \mathbf{b}^\star_{1,1} \| \cdots \| \mathbf{b}^\star_{1,\ell} \| \mathbf{a}^\star_{2,1} \| \cdots \| \mathbf{a}^\star_{2,\ell} \| \mathbf{b}^\star_{2,1} \| \cdots \| \mathbf{b}^\star_{2,\ell}).$$

Since $\tau(\mathsf{rdec}(p))$ has been included in $\mathbf{w}_2$, we now combine the remaining secret vectors appearing in equations (14), (16) into $\mathbf{w}_3 \in \{-1,0,1\}^{nm+4n\ell\delta_B}$ of the form

$$\mathbf{w}_3 = \left(\mathbf{x}^\star \parallel \mathbf{e}^\star_{1,1} \parallel \mathbf{e}^\star_{1,2} \parallel \mathbf{e}^\star_{2,1} \parallel \mathbf{e}^\star_{2,2} \right)$$

and $\mathbf{w}_4 \in \{-1,0,1\}^{4n^2\ell^2\delta_B}$ of the form

$$\mathbf{w}_4 = \big(\mathbf{y}_{1,1} \| \cdots \| \mathbf{y}_{1,\ell} \| \mathbf{z}_{1,1} \| \cdots \| \mathbf{z}_{1,\ell} \| \mathbf{y}_{2,1} \| \cdots \| \mathbf{y}_{2,\ell} \| \mathbf{z}_{2,1} \| \cdots \| \mathbf{z}_{2,\ell}\big)$$

such that for $i \in \{1,2\}$, and $j \in [\ell]$, $\mathbf{y}_{i,j}, \mathbf{z}_{i,j}$ satisfy the equations in (15).

For the sake of simplicity when defining our tailored set VALID and permutation Γ_η, we rearrange our secret vectors $\mathbf{w}_2, \mathbf{w}_3$ into vector $\overline{\mathbf{w}}_2 \in \{-1,0,1\}^{L'_2}$ of the form

$$\overline{\mathbf{w}}_2 = \big(\mathbf{s}^\star \| \mathbf{r}^\star \| \tau(\mathbf{y}) \| \tau(\mathsf{rdec}(p)) \| \mathbf{x}^\star \| \mathbf{e}^\star_{1,1} \| \mathbf{e}^\star_{1,2} \| \mathbf{e}^\star_{2,1} \| \mathbf{e}^\star_{2,2}\big).$$

and $\overline{\mathbf{w}}_3 \in \{-1,0,1\}^{4n\ell^2}$ of the form

$$\overline{\mathbf{w}}_3 = \big(\mathbf{a}^\star_{1,1} \| \cdots \| \mathbf{a}^\star_{1,\ell} \| \mathbf{b}^\star_{1,1} \| \cdots \| \mathbf{b}^\star_{1,\ell} \| \mathbf{a}^\star_{2,1} \| \cdots \| \mathbf{a}^\star_{2,\ell} \| \mathbf{b}^\star_{2,1} \| \cdots \| \mathbf{b}^\star_{2,\ell}\big)$$

with $L'_2 = 2n\overline{m}\delta_\beta + 2n\ell + nm + 4n\ell\delta_B$. Now we form our secret vector as $\mathbf{w}_0 = (\mathbf{w}_1 \| \overline{\mathbf{w}}_2 \| \overline{\mathbf{w}}_3 \| \mathbf{w}_4)$.

Second, we apply the extension and permutation techniques from Section 2.5 and Section 4.1 to our secret vectors $\mathbf{w}_0$. Let $\mathbf{w}'_1 = \mathsf{mix}(t, \mathbf{z}^\star) \in \{-1,0,1\}^{L_1}$ be the "mixing" vector obtained in equation (9), $\mathbf{w}'_2 = \mathsf{enc}(\overline{\mathbf{w}}_2) \in \{-1,0,1\}^{L_2}$ $\mathbf{w}'_3 = \mathsf{enc}(\overline{\mathbf{w}}_3) \in \{-1,0,1\}^{L_3}$, and $\mathbf{w}'_4 = \mathsf{Mult}(\mathbf{w}_4) \in \{-1,0,1\}^{L_4}$ be of the following form:

$$\begin{aligned}\big(&\mathsf{mult}(\mathbf{a}^\star_{1,1}, \mathbf{g}^\star_1) \| \cdots \| \mathsf{mult}(\mathbf{a}^\star_{1,\ell}, \mathbf{g}^\star_1) \| \mathsf{mult}(\mathbf{b}^\star_{1,1}, \mathbf{g}^\star_1) \| \cdots \| \mathsf{mult}(\mathbf{b}^\star_{1,\ell}, \mathbf{g}^\star_1) \| \\ &\mathsf{mult}(\mathbf{a}^\star_{2,1}, \mathbf{g}^\star_2) \| \cdots \| \mathsf{mult}(\mathbf{a}^\star_{2,\ell}, \mathbf{g}^\star_2) \| \mathsf{mult}(\mathbf{b}^\star_{2,1}, \mathbf{g}^\star_2) \| \cdots \| \mathsf{mult}(\mathbf{b}^\star_{2,\ell}, \mathbf{g}^\star_2) \big),\end{aligned}$$

Where $L_1 = 3k\delta_\beta + 6nk\delta_\beta c_d$, $L_2 = 3L'_2$, $L_3 = 12n\ell^2$, and $L_4 = 36n^2\ell^2\delta_B$. Denote $L = L_1 + L_2 + L_3 + L_4$. Form our extended vector $\mathbf{w} = (\mathbf{w}'_1 \| \mathbf{w}'_2 \| \mathbf{w}'_3 \| \mathbf{w}'_4) \in \{-1,0,1\}^L$.

Following the process in Section 2.6 and Section 4.2, we are able to obtain public matrix/vector $\mathbf{M}$ and $\mathbf{u}$ such that the considered statement is reduced to $\mathbf{M} \cdot \mathbf{w} = \mathbf{u} \bmod q$. Therefore, we are prepared to define the set VALID that includes our secret vector $\mathbf{w}$, the set $\mathcal{S}$, and the associated permutations $\{\Gamma_\eta : \eta \in \mathcal{S}\}$, such that the conditions in (1) are satisfied.

Let VALID be the set of all vectors $\mathbf{v}' = (\mathbf{v}'_1 \| \mathbf{v}'_2 \| \mathbf{v}'_3 \| \mathbf{v}'_4) \in \{-1,0,1\}^L$ such that the following requirements hold:

- $\mathbf{v}'_1 = \mathsf{mix}(t, \mathbf{z}^\star)$ for some $t \in \{0,1\}^{c_d}$ and $\mathbf{z}^\star \in \{-1,0,1\}^{nk\delta_\beta}$.
- $\mathbf{v}'_2 = \mathsf{enc}(\overline{\mathbf{w}}_2)$ for some $\overline{\mathbf{w}}_2 \in \{-1,0,1\}^{L'_2}$.
- For $j \in [4\ell]$, there exists $\overline{\mathbf{w}}_{3,j} \in \{-1,0,1\}^{n\ell}$ and $\overline{\mathbf{w}}_3 = (\overline{\mathbf{w}}_{3,1} \cdots \| \overline{\mathbf{w}}_{3,4\ell}) \in \{-1,0,1\}^{4n\ell^2}$ such that $\mathbf{v}'_3 = (\mathsf{enc}(\overline{\mathbf{w}}_{3,1}) \| \cdots \| \mathsf{enc}(\overline{\mathbf{w}}_{3,4\ell})) = \mathsf{enc}(\overline{\mathbf{w}}_3)$.

– There exists $\mathbf{g}_1^\star, \mathbf{g}_2^\star \in \{-1,0,1\}^{n\delta_B}$ and $\mathbf{w}_4 \in \{-1,0,1\}^{4n^2\ell^2\delta_B}$ be of the form:

$$(\mathsf{expd}(\overline{\mathbf{w}}_{3,1}, \mathbf{g}_1^\star) \| \cdots \| \mathsf{expd}(\overline{\mathbf{w}}_{3,2\ell}, \mathbf{g}_1^\star) \| \mathsf{expd}(\overline{\mathbf{w}}_{3,2\ell+1}, \mathbf{g}_2^\star) \| \cdots \| \mathsf{expd}(\overline{\mathbf{w}}_{3,4\ell}, \mathbf{g}_2^\star))$$

such that $\mathbf{v}_4' = \mathsf{Mult}(\mathbf{w}_4)$.

It is verifiable that our secret vector $\mathbf{w}$ belongs to VALID.

Now let $\mathcal{S} = \{0,1\}^{c_d} \times \{-1,0,1\}^{nk\delta_\beta} \times \{-1,0,1\}^{L_2'} \times (\{-1,0,1\}^{n\ell})^{4\ell} \times (\{-1,0,1\}^{n\delta_B})^2$, and associate every element

$$\eta = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_{4,1}, \ldots, \mathbf{f}_{4,2\ell}, \mathbf{f}_{5,1}, \ldots, \mathbf{f}_{5,2\ell}, \mathbf{f}_6, \mathbf{f}_7) \in \mathcal{S}$$

with Γ_η that works as follows. For a vector of form $\mathbf{v}^\star = (\mathbf{v}_1^\star \| \mathbf{v}_2^\star \| \mathbf{v}_3^\star \| \mathbf{v}_4^\star) \in \mathbb{Z}^L$, where $\mathbf{v}_i^\star \in \mathbb{Z}^{L_i}$ for $i \in \{1,2\}$, $\mathbf{v}_3^\star = (\mathbf{v}_{3,1}^\star \| \cdots \| \mathbf{v}_{3,4\ell}^\star)$ with $\mathbf{v}_{3,j}^\star \in \mathbb{Z}^{3n\ell}$, and $\mathbf{v}_4^\star = (\mathbf{v}_{4,1}^\star \cdots \| \mathbf{v}_{4,4\ell}^\star)$ with $\mathbf{v}_{4,j}^\star \in \mathbb{Z}^{9n^2\ell\delta_B}$, it transforms $\mathbf{v}^\star$ into vector $\Gamma_\eta(\mathbf{v}^\star)$

$$\begin{aligned}
&(\Psi_{\mathbf{f}_1,\mathbf{f}_2}(\mathbf{v}_1^\star) \,\|\, \Pi_{\mathbf{f}_3}(\mathbf{v}_2^\star) \,\| \\
&\;\Pi_{\mathbf{f}_{4,1}}(\mathbf{v}_{3,1}^\star) \,\|\, \cdots \,\|\, \Pi_{\mathbf{f}_{4,2\ell}}(\mathbf{v}_{3,2\ell}^\star) \,\|\, \Pi_{\mathbf{f}_{5,1}}(\mathbf{v}_{3,2\ell+1}^\star) \,\|\, \cdots \,\|\, \Pi_{\mathbf{f}_{5,2\ell}}(\mathbf{v}_{3,4\ell}^\star) \| \\
&\;\Phi_{\mathbf{f}_{4,1},\mathbf{f}_6}(\mathbf{v}_{4,1}^\star) \,\|\, \cdots \,\|\, \Phi_{\mathbf{f}_{4,2\ell},\mathbf{f}_6}(\mathbf{v}_{4,2\ell}^\star) \,\|\, \Phi_{\mathbf{f}_{5,1},\mathbf{f}_7}(\mathbf{v}_{4,2\ell+1}^\star) \,\|\, \cdots \,\|\, \Phi_{\mathbf{f}_{5,2\ell},\mathbf{f}_7}(\mathbf{v}_{4,4\ell}^\star)\,)
\end{aligned}$$

It then follows from the equivalences in (4), (6), and (11) that VALID, $\mathcal{S}$, and Γ_η satisfy the conditions in (1). Therefore, we have transformed the considered statement to a case of the abstract protocol from Section 2.4. To obtain the desired statistical ZKAoK protocol, it suffices for the prover and verifier to run the interactive protocol described in Figure 1. The protocol has perfect completeness, soundness error $2/3$ and communication cost $\mathcal{O}(L \cdot \log q)$, which is of the order $\mathcal{O}(n^2 \cdot \log^3 n) = \widetilde{\mathcal{O}}(\lambda^2)$.

5.2 Description of Our ATS Scheme

We assume there is a trusted setup such that it generates parameters of the scheme. Specifically, it generates a public matrix $\mathbf{B}$ for generating users' key pairs, and two secret-public key pairs of our KOE scheme such that the secret keys are discarded and not known by any party. The group public key then consists of three parts: (i) the parameters from the trusted setup, (ii) a verification key of the Ducas-Micciancio signature, (iii) two public keys of our KOE scheme such that the group manager knows both secret keys. The issue key is the Ducas-Micciancio signing key, while the opening key is any one of the corresponding secret keys of the two public keys. Note that both the issue key and the opening key are generated by the group manager.

When a user joins the group, it first generates a secret-public key pair $(\mathbf{x}, p)$ such that $\mathbf{B} \cdot \mathbf{x} = p$. It then interacts with the group manager, who will determine whether user p is traceable or not. If the user is traceable, group manager sets a bit $\mathsf{tr} = 1$, randomizes the two public key generated by himself, and then generates a Ducas-Micciancio signature σ_{cert} on user public key p and the two

randomized public keys $(\mathsf{epk}_1, \mathsf{epk}_2)$. If the user is non-traceable, group manager sets a bit $\mathsf{tr} = 0$, randomizes the two public key generated from the trusted setup, and then generates a signature on p and $\mathsf{epk}_1, \mathsf{epk}_2$. If it completes successfully, the group manager sends certificate $\mathsf{cert} = (p, \mathsf{epk}_1, \mathsf{epk}_2, \sigma_{\mathsf{cert}})$ to user p, registers this user to the group, and keeps himself the witness w^{escrw} that was ever used for randomization.

Once registered as a group member, the user can sign messages on behalf of the group. To this end, the user first encrypts his public key p twice using his two randomized public keys, and obtains ciphertexts $\mathbf{c}_1, \mathbf{c}_2$. The user then generates a ZKAoK such that (i) he has a valid secret key $\mathbf{x}$ corresponding to p; (ii) he possesses a Ducas-Micciancio signature on p and $\mathsf{epk}_1, \mathsf{epk}_2$; and (iii) $\mathbf{c}_1, \mathbf{c}_2$ are correct ciphertexts of p under the randomized keys $\mathsf{epk}_1, \mathsf{epk}_2$, respectively. Since the ZKAoK protocol the user employs has soundness error $2/3$ in each execution, it is repeated $\kappa = \omega(\log \lambda)$ times to make the error negligibly small. Then, it is made non-interactive via the Fiat-Shamir heuristic [17]. The signature then consists of the non-interactive zero-knowledge argument of knowledge ($\mathsf{NIZKAoK}$) Π_{gs} and the two ciphertexts. Note that the ZK argument together with double encryption enables CCA-security of the underlying encryption scheme, which is known as the Naor-Yung transformation [47].

To verify the validity of a signature, it suffices to verify the validity of the argument Π_{gs}. Should the need arises, the group manager can decrypt using his opening key. If a user is traceable, the opening key group manager possesses can be used to correctly identify the signer. However, if a user is non-traceable, then his anonymity is preserved against the manager.

To prevent corrupted opening, group manager is required to generate a $\mathsf{NIZKAoK}$ of correct opening Π_{open}. Only when Π_{open} is a valid argument, we then accept the opening result. Furthermore, there is an additional accounting mechanism for group manager to reveal which users he had chosen to be traceable. This is done by checking the consistency of tr and the randomized public keys in user's certificate with the help of the witness w^{escrw}.

We describe the details of our scheme below.

$\mathsf{Setup}(\lambda)$**:** Given the security parameter λ, it generates the following public parameter.

- Let $n = \mathcal{O}(\lambda)$ be a power of 2, and modulus $q = \widetilde{\mathcal{O}}(n^4)$, where $q = 3^k$ for $k \in \mathbb{Z}^+$. Let $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = R/qR$.
  Also, let $m \geq 2\lceil \log q \rceil + 2$, $\ell = \lfloor \log \frac{q-1}{2} \rfloor + 1$, $m_s = 4\ell + 1$, and $\overline{m} = m + k$ and $\overline{m}_s = m_s \cdot \ell$.
- Let integer d and sequence $c_0, \ldots, c_d$ be described in Section 2.3.
- Let $\beta = \widetilde{\mathcal{O}}(n)$ and $B = \widetilde{\mathcal{O}}(\sqrt{n})$ be two integer bounds, and χ be a B-bounded distribution over the ring R.
- Choose a collision-resistant hash function $\mathcal{H}_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^\kappa$, where $\kappa = \omega(\log \lambda)$, which will act as a random oracle in the Fiat-Shamir heuristic [17].
- Choose a statistically hiding and computationally binding commitment scheme from [22], denoted as COM, which will be employed in our ZK argument systems.

– Let $\mathbf{B} \xleftarrow{\$} R_q^{1\times m}$, $\mathbf{a}_1^{(0)} \xleftarrow{\$} R_q^{\ell}$, $\mathbf{a}_2^{(0)} \xleftarrow{\$} R_q^{\ell}$, $s_{-1}, s_{-2} \hookleftarrow \chi$, $\mathbf{e}_{-1}, \mathbf{e}_{-2} \hookleftarrow \chi^{\ell}$. Compute

$$\mathbf{b}_1^{(0)} = \mathbf{a}_1^{(0)} \cdot s_{-1} + \mathbf{e}_{-1} \in R_q^{\ell}; \quad \mathbf{b}_2^{(0)} = \mathbf{a}_2^{(0)} \cdot s_{-2} + \mathbf{e}_{-2} \in R_q^{\ell}.$$

This algorithm outputs the public parameter pp:

$$\{\ \ n, q, k, R, R_q, \ell, m, m_s, \overline{m}, \overline{m}_s, d, c_0, \cdots, c_d,$$
$$\beta, B, \chi, \mathcal{H}_{\mathsf{FS}}, \kappa, \mathsf{COM}, \mathbf{B}, \{\mathbf{a}_i^{(0)}, \mathbf{b}_i^{(0)}\}_{i\in\{1,2\}} \ \ \}.$$

pp is implicit for all algorithms below if not explicitly mentioned.

$\mathsf{GKeyGen}(\mathsf{pp})$**:** On input pp, GM proceeds as follows.

– Generate verification key

$$\mathbf{A}, \mathbf{F}_0 \in R_q^{1\times\overline{m}};\ \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]} \in R_q^{1\times k};\ \mathbf{F} \in R_q^{1\times\ell};\ \mathbf{F}_1 \in R_q^{1\times\overline{m}_s};\ u \in R_q$$

and signing key $\mathbf{R} \in R_q^{m\times k}$ for the Ducas-Micciancio signature from Section 2.3.

– Initialize the Naor-Yung double-encryption mechanism [47] with the key-oblivious encryption scheme described in Section 3.1. Specifically, sample $s_1, s_2 \hookleftarrow \chi$, $\mathbf{e}_1, \mathbf{e}_2 \hookleftarrow \chi^{\ell}$, $\mathbf{a}_1^{(1)} \xleftarrow{\$} R_q^{\ell}$, $\mathbf{a}_2^{(1)} \xleftarrow{\$} R_q^{\ell}$ and compute

$$\mathbf{b}_1^{(1)} = \mathbf{a}_1^{(1)} \cdot s_1 + \mathbf{e}_1 \in R_q^{\ell}; \quad \mathbf{b}_2^{(1)} = \mathbf{a}_2^{(1)} \cdot s_2 + \mathbf{e}_2 \in R_q^{\ell}.$$

Set the group public key gpk, the issue key ik and the opening key ok as follows:

$$\mathsf{gpk} = \{\mathsf{pp}, \mathbf{A}, \{\mathbf{A}_{[j]}\}_{j=0}^{d}, \mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, u, \mathbf{a}_1^{(1)}, \mathbf{b}_1^{(1)}, \mathbf{a}_2^{(1)}, \mathbf{b}_2^{(1)} \ \ \},$$
$$\mathsf{ik} = \mathbf{R}, \quad \mathsf{ok} = (s_1, \mathbf{e}_1).$$

GM then makes gpk public, sets the registration table $\mathbf{reg} = \emptyset$ and his internal state $S = 0$.

$\mathsf{UKeyGen}(\mathsf{pp})$**:** Given the public parameter, the user first chooses $\mathbf{x} \in R^m$ such that the coefficients are uniformly chosen from the set $\{-1, 0, 1\}$. He then calculates $p = \mathbf{B} \cdot \mathbf{x} \in R_q$. Set $\mathsf{upk} = p$ and $\mathsf{usk} = \mathbf{x}$.

$\mathsf{Enroll}(\mathsf{gpk}, \mathsf{ik}, \mathsf{upk}, \mathsf{tr})$**:** Upon receiving a user public key upk from a user, GM determines the value of the bit $\mathsf{tr} \in \{0, 1\}$, indicating whether the user is traceable. He then does the following:

– Randomize two pairs of public keys $(\mathbf{a}_1^{(\mathsf{tr})}, \mathbf{b}_1^{(\mathsf{tr})})$ and $(\mathbf{a}_2^{(\mathsf{tr})}, \mathbf{b}_2^{(\mathsf{tr})})$ as described in Section 3.1. Specifically, sample $g_1, g_2 \hookleftarrow \chi$, $\mathbf{e}_{1,1}, \mathbf{e}_{1,2} \hookleftarrow \chi^{\ell}$, $\mathbf{e}_{2,1}, \mathbf{e}_{2,2} \hookleftarrow \chi^{\ell}$. For each $i \in \{1, 2\}$, compute

$$\mathsf{epk}_i = (\mathbf{a}_i', \mathbf{b}_i') = (\mathbf{a}_i^{(\mathsf{tr})} \cdot g_i + \mathbf{e}_{i,1}, \ \ \mathbf{b}_i^{(\mathsf{tr})} \cdot g_i + \mathbf{e}_{i,2}) \in R_q^{\ell} \times R_q^{\ell}. \quad (17)$$

– Set the tag $t = (t_0, t_1 \ldots, t_{c_d-1})^{\top} \in \mathcal{T}_d$, where $S = \sum_{j=0}^{c_d-1} 2^j \cdot t_j$, and compute $\mathbf{A}_t = [\mathbf{A} | \mathbf{A}_{[0]} + \sum_{i=1}^{d} t_{[i]} \mathbf{A}_{[i]}] \in R_q^{1\times(\overline{m}+k)}$.

– Let $\mathfrak{m} = (p\|\mathbf{a}'_1\|\mathbf{b}'_1\|\mathbf{a}'_2\|\mathbf{b}'_2) \in R_q^{m_s}$.

– Generate a signature $\sigma_{\mathsf{cert}} = (t, \mathbf{r}, \mathbf{v})$ on message $\mathsf{rdec}(\mathfrak{m}) \in R^{\overline{m}_s}$ - whose coefficients are in $\{-1, 0, 1\}$ - using his issue key $\mathsf{ik} = \mathbf{R}$. As in Section 2.3, we have $\mathbf{r} \in R^{\overline{m}}$, $\mathbf{v} \in R^{\overline{m}+k}$ and

$$\begin{cases} \mathbf{A}_t \cdot \mathbf{v} = \mathbf{F} \cdot \mathsf{rdec}(\mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \mathsf{rdec}(\mathfrak{m})) + u, \\ \|\mathbf{r}\|_\infty \leq \beta, \quad \|\mathbf{v}\|_\infty \leq \beta. \end{cases} \tag{18}$$

Set certificate cert and w^{escrw} as follows:

$$\mathsf{cert} = (p, \mathbf{a}'_1, \mathbf{b}'_1, \mathbf{a}'_2, \mathbf{b}'_2, t, \mathbf{r}, \mathbf{v}), \quad w^{\mathsf{escrw}} = (g_1, \mathbf{e}_{1,1}, \mathbf{e}_{1,2}, g_2, \mathbf{e}_{2,1}, \mathbf{e}_{2,2}).$$

GM sends cert to the user p, stores $\mathbf{reg}[S] = (p, \mathsf{tr}, w^{\mathsf{escrw}})$, and updates the state to $S + 1$.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}, \mathsf{usk}, M)$**:** To sign a message $M \in \{0,1\}^*$ using the certificate $\mathsf{cert} = (p, \mathbf{a}'_1, \mathbf{b}'_1, \mathbf{a}'_2, \mathbf{b}'_2, t, \mathbf{r}, \mathbf{v})$ and $\mathsf{usk} = \mathbf{x}$, the user proceeds as follows.

– Encrypt the ring vector $\mathsf{rdec}(p) \in R_q^\ell$ whose coefficients are in $\{-1, 0, 1\}$ twice. Namely, sample $g'_1, g'_2 \hookleftarrow \chi$, $\mathbf{e}'_{1,1}, \mathbf{e}'_{1,2} \hookleftarrow \chi^\ell$, and $\mathbf{e}'_{2,1}, \mathbf{e}'_{2,2} \hookleftarrow \chi^\ell$. For each $i \in \{1, 2\}$, compute $\mathbf{c}_i = (\mathbf{c}_{i,1}, \mathbf{c}_{i,2}) \in R_q^\ell \times R_q^\ell$ as follows:

$$\mathbf{c}_{i,1} = \mathbf{a}'_i \cdot g'_i + \mathbf{e}'_{i,1}; \quad \mathbf{c}_{i,2} = \mathbf{b}'_i \cdot g'_i + \mathbf{e}'_{i,2} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p).$$

– Generate a $\mathsf{NIZKAoK}$ Π_{gs} to demonstrate the possession of a valid tuple ζ of the following form

$$\zeta = (p, \mathbf{a}'_1, \mathbf{b}'_1, \mathbf{a}'_2, \mathbf{b}'_2, t, \mathbf{r}, \mathbf{v}, \mathbf{x}, g'_1, \mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}, g'_2, \mathbf{e}'_{2,1}, \mathbf{e}'_{2,2}) \tag{19}$$

such that

(i) The conditions in (18) are satisfied.

(ii) $\mathbf{c}_1$ and $\mathbf{c}_2$ are correct encryptions of $\mathsf{rdec}(p)$ with B-bounded randomness $g'_1, \mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}$ and $g'_2, \mathbf{e}'_{2,1}, \mathbf{e}'_{2,2}$, respectively.

(iii) $\|\mathbf{x}\|_\infty \leq 1$ and $\mathbf{B} \cdot \mathbf{x} = p$.

This is achieved by running the protocol from Section 5.1, which is repeated $\kappa = \omega(\log \lambda)$ times and made non-interactive via Fiat-Shamir heuristic [17] as a triple $\Pi_{\mathsf{gs}} = (\{\mathrm{CMT}_i\}_{i=1}^\kappa, \mathrm{CH}, \{\mathrm{RSP}_i\}_{i=1}^\kappa)$ where the challenge CH is generated as $\mathrm{CH} = \mathcal{H}_{\mathsf{FS}}(M, \{\mathrm{CMT}_i\}_{i=1}^\kappa, \xi)$ with ξ of the following form

$$\xi = (\mathbf{A}, \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]}, \mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, u, \mathbf{B}, \mathbf{c}_1, \mathbf{c}_2) \tag{20}$$

– Output the group signature $\Sigma = (\Pi_{\mathsf{gs}}, \mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$**:** Given the inputs, the verifier performs in the following manner.

– Parse Σ as $\Sigma = \big(\{\mathrm{CMT}_i\}_{i=1}^\kappa, (Ch_1, \ldots, Ch_\kappa), \{\mathrm{RSP}\}_{i=1}^\kappa, \mathbf{c}_1, \mathbf{c}_2\big)$. If $(Ch_1, \ldots, Ch_\kappa) \neq \mathcal{H}_{\mathsf{FS}}\big(M, \{\mathrm{CMT}_i\}_{i=1}^\kappa, \xi\big)$, output 0, where ξ is as in (20).

– For each $i \in [\kappa]$, run the verification phase of the protocol in Section 5.1 to verify the validity of RSP_i corresponding to CMT_i and Ch_i. If any of the verification process fails, output 0.
– Output 1.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, M, \Sigma)$**:** Let $\mathsf{ok} = (s_1, \mathbf{e}_1)$ and $\Sigma = (\Pi_{\mathsf{gs}}, \mathbf{c}_1, \mathbf{c}_2)$. The group manager proceeds as follows.
– Use s_1 to decrypt $\mathbf{c}_1 = (\mathbf{c}_{1,1}, \mathbf{c}_{1,2})$ as in the decryption algorithm from Section 3.1. The result is $p' \in R_q$.
– He then searches the registration information. If **reg** does not include an element p', then return $\perp$.
– Otherwise, he produces a $\mathsf{NIZKAoK}$ Π_{open} to show the knowledge of a tuple $(s_1, \mathbf{e}_1, \mathbf{y}) \in R_q \times R_q^{\ell} \times R_q^{\ell}$ such that the following conditions hold.

$$\begin{cases} \|s_1\|_\infty \leq B;\ \|\mathbf{e}_1\|_\infty \leq B;\ \|\mathbf{y}\|_\infty \leq \lceil q/10 \rceil; \\ \mathbf{a}_1^{(1)} \cdot s_1 + \mathbf{e}_1 = \mathbf{b}_1^{(1)}; \\ \mathbf{c}_{1,2} - \mathbf{c}_{1,1} \cdot s_1 = \mathbf{y} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p'). \end{cases} \tag{21}$$

Since the conditions in (21) only encounter linear secret objects with bounded norm, we can easily handled them using the Stern-like techniques from Sections 4.2 and 5.1. Therefore, we are able to have a statistical ZKAoK for the above statement. Furthermore, the protocol is repeated $\kappa = \omega(\log \lambda)$ times and made non-interactive via the Fiat-Shamir heuristic, resulting in a triple $\Pi_{\mathsf{Open}} = (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathrm{CH}, \{\mathrm{RSP}\}_{i=1}^{\kappa})$, where $\mathrm{CH} \in \{1,2,3\}^{\kappa}$ is computed as

$$\mathrm{CH} = \mathcal{H}_{\mathsf{FS}}\big(\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathbf{a}_1^{(1)}, \mathbf{b}_1^{(1)}, M, \Sigma, p'\big). \tag{22}$$

– Output $(p', \Pi_{\mathsf{Open}})$.

$\mathsf{Judge}(\mathsf{gpk}, M, \Sigma, p', \Pi_{\mathsf{open}})$**:** Given all the inputs, this algorithm does the following.
– If Verify algorithm outputs 0 or $p' = \perp$, return 0.
– This algorithm then verifies the argument Π_{Open} with respect to common input $(\mathbf{a}_1^{(1)}, \mathbf{b}_1^{(1)}, M, \Sigma, p')$, in the same way as in the algorithm Verify. If verification of the argument Π_{open} fails, output 0.
– Else output 1.

$\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}, w^{\mathsf{escrw}}, \mathsf{tr})$**:** Let the certificate be $\mathsf{cert} = (p, \mathbf{a}_1', \mathbf{b}_1', \mathbf{a}_2', \mathbf{b}_2', t, \mathbf{r}, \mathbf{v})$ and witness be $w^{\mathsf{escrw}} = (g_1, \mathbf{e}_{1,1}, \mathbf{e}_{1,2}, g_2, \mathbf{e}_{2,1}, \mathbf{e}_{2,2})$ and the bit tr, this algorithm proceeds as follows.
– It checks whether $(t, \mathbf{r}, \mathbf{v})$ is a valid Ducas-Micciancio signature on the message $(p, \mathbf{a}_1', \mathbf{b}_1', \mathbf{a}_2', \mathbf{b}_2')$. Specifically, it verifies whether cert satisfies the conditions in (18). If not, output 0.
– Otherwise, it then checks if $(\mathbf{a}_1', \mathbf{b}_1')$ and $(\mathbf{a}_2', \mathbf{b}_2')$ are randomization of $(\mathbf{a}_1^{(\mathsf{tr})}, (\mathbf{b}_1^{(\mathsf{tr})})$ and $(\mathbf{a}_2^{(\mathsf{tr})}, (\mathbf{b}_2^{(\mathsf{tr})})$ with respect to randomness $(g_1, \mathbf{e}_{1,1}, \mathbf{e}_{1,2})$ and $(g_2, \mathbf{e}_{2,1}, \mathbf{e}_{2,2})$, respectively. Specifically, it verifies whether the conditions in (17) hold. If not, output 0.
– Else output 1.

5.3 Analysis of Our ATS Scheme

Efficiency. We first analyze the efficiency of our scheme from Section 5.2 in terms of the security parameter λ.

- The bit-size of the public key gpk is of order $\mathcal{O}(\lambda \cdot \log^3 \lambda) = \widetilde{\mathcal{O}}(\lambda)$.
- The bit-size of the membership certificate cert is of order $\mathcal{O}(\lambda \cdot \log^2 \lambda) = \widetilde{\mathcal{O}}(\lambda)$.
- The bit-size of a signature Σ is determined by that of the Stern-like $\mathsf{NIZKAoK}$ Π_{gs}, which is of order $\mathcal{O}(L \cdot \log q) \cdot \omega(\log \lambda)$, where L is the bit-size of a vector $\mathbf{w} \in \mathsf{VALID}$ from Section 5.1. Recall $\mathcal{O}(L \cdot \log q) = \mathcal{O}(\lambda^2 \cdot \log^3 \lambda)$. Therefore, the bit-size of Σ is of order $\mathcal{O}(\lambda^2 \cdot \log^3 \lambda) \cdot \omega(\log \lambda) = \widetilde{\mathcal{O}}(\lambda^2)$.
- The bit-size of the Stern-like $\mathsf{NIZKAoK}$ Π_{open} is of order $\mathcal{O}(\lambda \cdot \log^3 \lambda) \cdot \omega(\log \lambda) = \widetilde{\mathcal{O}}(\lambda)$.

Correctness. For an honestly generated signature Σ for message M, we first show that the Verify algorithm always outputs 1. Due to the honest behavior of the user, when signing a message in the name of the group, this user possesses a valid tuple ζ of the form (19). Therefore, Π_{gs} will be accepted by the Verify algorithm with probability 1 due to the perfect completeness of our argument system.

If an honest user is traceable, then $\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}, w^{\mathsf{escrw}}, 1)$ will output 1, implied by the correctness of Ducas-Micciancio signature scheme and honest behaviour of group manager. In terms of the correctness of the Open algorithm, we observe that $\mathbf{c}_{1,2} - \mathbf{c}_{1,1} \cdot s_1 =$

$$(\mathbf{b}_1^{(\mathsf{tr})} - \mathbf{a}_1^{(\mathsf{tr})} \cdot s_1) \cdot g_1 \cdot g_1' + \mathbf{e}_{1,2} \cdot g_1' - \mathbf{e}_{1,1} \cdot s_1 \cdot g_1' + \mathbf{e}_{1,2}' - \mathbf{e}_{1,1}' \cdot s_1 + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p),$$

denoted as $\widetilde{\mathbf{e}} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p)$. In this case, $\mathsf{tr} = 1$, $\mathbf{b}_1^{(\mathsf{tr})} - \mathbf{a}_1^{(\mathsf{tr})} \cdot s_1 = \mathbf{e}_1$, and $\|\widetilde{\mathbf{e}}\|_\infty \leq \lceil \frac{q}{10} \rceil$. The decryption can recover $\mathsf{rdec}(p)$ and hence the real signer due to the correctness of our key-oblivious encryption from Section 3.1. Thus, correctness of the Open algorithm follows. What is more, Π_{open} will be accepted by the Judge algorithm with probability 1 due to the perfect completeness of our argument system.

If an honest user is non-traceable, then again $\mathsf{Account}(\mathsf{gpk}, \mathsf{cert}, w^{\mathsf{escrw}}, 1)$ will output 1. For the Open algorithm, since $\mathbf{b}_1^{(0)} - \mathbf{a}_1^{(0)} \cdot s_1 = \mathbf{a}_1^{(0)} \cdot (s_{-1} - s_1) + \mathbf{e}_{-1}$, then we obtain

$$\mathbf{c}_{1,2} - \mathbf{c}_{1,1} \cdot s_1 = \mathbf{a}_1^{(0)} \cdot (s_{-1} - s_1) \cdot g_1 \cdot g_1' + \widetilde{\mathbf{e}} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p),$$

where $\|\widetilde{\mathbf{e}}\|_\infty \leq \lceil \frac{q}{10} \rceil$. Observe that $\mathbf{a}_1^{(0)} \xleftarrow{\$} R_q^\ell$, and $s_{-1} \neq s_1$ with overwhelming probability. Over the randomness of g_1, g_1', the decryption algorithm described in Section 3.1 will output a random element $p' \in R_q$. Then, with overwhelming probability, p' is not in the registration table and the Open algorithm outputs $\perp$. It then follows that our scheme is correct.

Security. In Theorem 3, we prove that our scheme satisfies the security requirements of accountable tracing signatures, as specified by Kohlweiss and Miers.

Theorem 3. *Under the* RLWE *and* RSIS *assumptions, the accountable tracing signature scheme described in Section 5.2 satisfies the following requirements in the random oracle model: (i) anonymity under tracing; (ii) traceability; (iii) non-frameability; (iv) anonymity with accountability; and (v) trace-obliviousness.*

For the proofs of traceability and non-frameability, the lemma below from [36] is needed.

Lemma 5 ([36]). *Let* $\mathbf{B} \in R_q^{1\times m}$*, where* $m \geq 2\lceil \log q \rceil + 2$*. If* $\mathbf{x}$ *is a uniform element over* R^m *with* $\|\mathbf{x}\|_\infty \leq 1$*, then with probability at least* $1 - 2^{-n}$*, there exists a different* $\mathbf{x}' \in R^m$ *with* $\|\mathbf{x}'\|_\infty \leq 1$ *and* $\mathbf{B} \cdot \mathbf{x}' = \mathbf{B} \cdot \mathbf{x} \in R_q$*.*

The proof of the Theorem 3 follows from Lemma 6-10 given below.

Lemma 6. *Assuming the hardness of the* RLWE *problem, in the random oracle model, the given accountable tracing signature scheme is anonymous under tracing.*

Proof. We prove this lemma using a series of indistinguishable games. In the initial game, the challenger runs the experiment $\mathbf{Exp}^{\mathsf{AuT}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda)$ while in the last game, the challenger runs the experiment $\mathbf{Exp}^{\mathsf{AuT}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda)$. Let W_i be the event that the adversary outputs 1 in Game i.

Game 0**:** This is exactly the experiment $\mathbf{Exp}^{\mathsf{AuT}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda)$, where the adversary receives a challenged signature $(\Pi^*_{\mathsf{gs}}, \mathbf{c}^*_1, \mathbf{c}^*_2) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}_0, \mathsf{usk}_0, M)$ in the challenge phase with $p_0 = \mathbf{B} \cdot \mathsf{usk}_0$. So $\Pr[W_0] = \Pr[\mathbf{Exp}^{\mathsf{AuT}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$.

Game 1**:** We modify Game 0 as follows: the challenger will keep decryption key $(s_2, \mathbf{e}_2)$ secret (by himself) instead of erasing it. However, the view of the adversary $\mathcal{A}$ is still the same as in Game 0. Therefore, $\Pr[W_0] = \Pr[W_1]$.

Game 2**:** This game is the same as Game 1 with one exception: it generates simulated proofs for the opening oracle queries by programming the random oracle $\mathcal{H}_{\mathsf{FS}}$. Note that the challenger still follows the original game (that is, it uses s_1 to decrypt $\mathbf{c}_1$) to identify the real signer. The views of $\mathcal{A}$ in Game 1 and Game 2 are statistically close due to the statistical zero-knowledge property of our argument system. Therefore $\Pr[W_1] \overset{s}{\approx} \Pr[W_2]$.

Game 3**:** This game modifies Game 2 as follows. It uses s_2 instead of s_1 to answer the opening oracle queries. In other words, it now uses $\mathbf{s}_2$ to decrypt $\mathbf{c}_2$ to identify the signer. The view of the adversary in this game is identical to that in Game 2 until event F_1, where $\mathcal{A}$ queries the opening oracle a valid signature $(\Pi_{\mathsf{gs}}, \mathbf{c}_1, \mathbf{c}_2)$ with $\mathbf{c}_1, \mathbf{c}_2$ encrypting distinct messages, happens. Since the event F_1 violates the soundness of our argument system, we have $|\Pr[W_2] - \Pr[W_3]| \leq \Pr[F_1] \leq \mathbf{Adv}^{\mathsf{sound}}_{\Pi_{\mathsf{gs}}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 4**:** This game changes Game 3 as follows. It generates a simulated proof Π^*_{gs} in the challenge phase even though the challenger has the correct witness to generate a real proof. Due to the statistical zero-knowledge property of our argument system, this change is negligible to $\mathcal{A}$. Therefore $\Pr[W_3] \overset{s}{\approx} \Pr[W_4]$.

Game 5**:** In this game, we modify Game 4 by modifying the distribution of the challenged signature $\Sigma^* = (\Pi^*_{\mathsf{gs}}, \mathbf{c}^*_1, \mathbf{c}^*_2)$ as follows. For $i \in \{0,1\}$, parse $\mathsf{cert}_i = (p_i, \mathbf{a}'_{1,i}, \mathbf{b}'_{1,i}, \mathbf{a}'_{2,i}, \mathbf{b}'_{2,i}, t_i, \mathbf{r}_i, \mathbf{v}_i)$. Recall that in Game 4, both $\mathbf{c}^*_1$ and $\mathbf{c}^*_2$ encrypt the same message, i.e., $\mathsf{rdec}(p_0)$, under the randomized key $(\mathbf{a}'_{1,0}, \mathbf{b}'_{1,0})$ and $(\mathbf{a}'_{2,0}, \mathbf{b}'_{2,0})$, respectively. Here we change $\mathbf{c}^*_1$ to be encryption of $\mathsf{rdec}(p_1)$ and keep $\mathbf{c}^*_2$ unchanged. By the semantic security under key randomization of our key oblivious encryption scheme for public key $(\mathbf{a}^{(1)}_1, \mathbf{b}^{(1)}_1)$ (which is implied by the RLWE assumption since we no longer use s_1 to open signatures), the change made in this game is negligible to the adversary. Therefore we have $|\Pr[W_4] - \Pr[W_5]| = \mathsf{negl}(\lambda)$.

Game 6**:** In this game, we further modify the distribution of the challenged signature Σ^*. We change $\mathbf{c}^*_1$ to be encryption of $\mathsf{rdec}(p_1)$ under a fresh and then randomized key. By the property of key privacy under key randomization of our key-oblivious encryption scheme, the change made in this game is negligible to the adversary. Therefore we have $|\Pr[W_5] - \Pr[W_6]| = \mathsf{negl}(\lambda)$.

Game 7**:** In this game, we again modify the distribution of the challenged signature Σ^*. We change $\mathbf{c}^*_1$ to be encryption of $\mathsf{rdec}(p_1)$ under the randomized key $(\mathbf{a}'_{1,1}, \mathbf{b}'_{1,1})$. By the same argument of indistinguishability between Game 6 and Game 5, we have $|\Pr[W_6] - \Pr[W_7]| = \mathsf{negl}(\lambda)$.

Game 8**:** This game is the same as Game 7 with one modification: it changes back to s_1 for the opening oracle queries and erases $(s_2, \mathbf{e}_2)$ again. This change is indistinguishable to $\mathcal{A}$ until event F_2, where $\mathcal{A}$ queries a valid signature $(\Pi_{\mathsf{gs}}, \mathbf{c}_1, \mathbf{c}_2)$ with $\mathbf{c}_1, \mathbf{c}_2$ encrypting different messages to the opening oracle, occurs. Since event F_2 violates the simulation soundness of our argument system, we have $|\Pr[W_7] - \Pr[W_8]| \leq \mathsf{Adv}^{\mathsf{ss}}_{\Pi_{\mathsf{gs}}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 9**:** In this game, we modify Game 8 by modifying the distribution of the challenged signature $\Sigma^* = (\Pi^*_{\mathsf{gs}}, \mathbf{c}^*_1, \mathbf{c}^*_2)$ again. It changes $\mathbf{c}^*_2$ to be encryption of $\mathsf{rdec}(p_1)$ under the randomized key $(\mathbf{a}'_{2,1}, \mathbf{b}'_{2,1})$ in the challenge phase. By the same argument of indistinguishability from Game 4 to Game 7, we have $|\Pr[W_8] - \Pr[W_9]| = \mathsf{negl}(\lambda)$.

Game 10**:** Note that in Game 9, both $\mathbf{c}^*_1$ and $\mathbf{c}^*_2$ encrypt the same message, i.e., $\mathsf{rdec}(p_1)$, under the randomized key $(\mathbf{a}'_{1,1}, \mathbf{b}'_{1,1})$ and $(\mathbf{a}'_{2,1}, \mathbf{b}'_{2,1})$, respectively. Therefore, the challenger has correct witness to generate Π^*_{gs}. In this game, we modify Game 9 by switching back to a real proof Π^*_{gs} in the challenge phase. Then the views of $\mathcal{A}$ in Game 9 and Game 10 are statistically indistinguishable by the statistical zero-knowledge property of our argument system. Hence $\Pr[W_9] \overset{s}{\approx} \Pr[W_{10}]$.

Game 11**:** This game changes Game 10 in one aspect. It now generates real proofs for the opening oracle queries. Due to the statistical zero-knowledge property of our argument system, Game 10 and Game 11 are statistically indistinguishable to $\mathcal{A}$. In other words, we have $\Pr[W_{10}] \overset{s}{\approx} \Pr[W_{11}]$. This is indeed the experiment $\mathbf{Exp}^{\mathsf{AuT}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda)$. Hence, we have $\Pr[W_{11}] = \Pr[\mathbf{Exp}^{\mathsf{AuT}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$.

As a result, we obtain

$$|\Pr[\mathbf{Exp}^{\mathsf{AuT}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda)=1]-\Pr[\mathbf{Exp}^{\mathsf{AuT}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda)=\ 1]|=\mathsf{negl}(\lambda),$$

and hence our scheme is anonymous under tracing.

Lemma 7. *Assuming the hardness of the* RSIS *problem, in the random oracle model, the given accountable tracing signature scheme is traceable .*

Proof. We show that the success probability ϵ of $\mathcal{A}$ against traceability is negligible by the unforgeability of the Ducas-Micciancio signature recalled in Section 2.3, which in turn relies on the hardness of the RSIS problem, or by the hardness of solving a RSIS instance directly.

Let $\mathcal{C}$ be the challenger and honestly run the experiment $\mathbf{Exp}^{\mathsf{Trace}}_{\mathsf{ATS},\mathcal{A}}(\lambda)$. When $\mathcal{A}$ halts, it outputs $(M^*, \mathit{\Pi}^*_{\mathsf{gs}}, \mathbf{c}^*_1, \mathbf{c}^*_2)$. Let us consider the case that $\mathcal{A}$ wins. Parse $\mathit{\Pi}^*_{\mathsf{gs}} = (\{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \mathrm{CH}^*, \{\mathrm{RSP}^*_i\}^{\kappa}_{i=1})$. Let

$$\xi^* = (\mathbf{A}, \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]}, \mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, u, \mathbf{B}, \mathbf{c}^*_1, \mathbf{c}^*_2).$$

Then $\mathrm{CH}^* = \mathcal{H}_{\mathsf{FS}}\big(M^*, \{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \xi^*\big)$ and for each $i \in [\kappa]$, RSP^*_i is a valid response corresponding to CMT^*_i and CH^*_i. This is due to the fact that $\mathcal{A}$ wins and hence $\mathit{\Pi}^*_{\mathsf{gs}}$ passes the verification process.

We remark that $\mathcal{A}$ had queried the tuple $\big(M^*, \{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \xi^*\big)$ to the hash oracle $\mathcal{H}_{\mathsf{FS}}$ with all but negligible probability. Since we can only guess correctly the value $\mathcal{H}_{\mathsf{FS}}\big(M^*, \{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \xi^*\big)$ with probability $3^{-\kappa}$, which is negligible. Therefore, $\mathcal{A}$ had queried the tuple $\big(M^*, \{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \xi^*\big)$ to $\mathcal{H}_{\mathsf{FS}}$ with probability $\epsilon' = \epsilon - 3^{-\kappa}$. Let this tuple be the θ^*-th oracle query made by $\mathcal{A}$ and assume $\mathcal{A}$ had made Q_H queries in total.

Up to this point, the challenger $\mathcal{C}$ then replays the behaviour of $\mathcal{A}$ for at most $32 \cdot Q_H/\epsilon'$ times. In each new replay, $\mathcal{A}$ is given the same hash answers $r_1, \ldots, r_{\theta^*-1}$ as in the original run for the first θ^*-1 hash queries while it is given uniformly random and independent values $r'_{\theta^*}, \ldots, r'_{Q_H}$ for the remaining hash queries. According to the forking lemma of Brickell et al. [11], with probability $\geq 1/2$, $\mathcal{B}$ obtains 3-fork involving the same tuple $\big(M^*, \{\mathrm{CMT}^*_i\}^{\kappa}_{i=1}, \xi^*\big)$ with pairwise distinct hash values $\mathrm{CH}^{(1)}_{\theta^*}, \mathrm{CH}^{(2)}_{\theta^*}, \mathrm{CH}^{(3)}_{\theta^*} \in \{1,2,3\}^{\kappa}$ and corresponding valid responses $\mathrm{RSP}^{(1)}_{\theta^*}$, $\mathrm{RSP}^{(2)}_{\theta^*}$, $\mathrm{RSP}^{(3)}_{\theta^*}$. We observe that with probability $1 - (\frac{7}{9})^{\kappa}$, there exists some $j \in \{1, 2, \ldots, \kappa\}$ such that $\{\mathsf{CH}^{(1)}_{\theta^*,j}, \mathsf{CH}^{(2)}_{\theta^*,j}, \mathsf{CH}^{(3)}_{\theta^*,j}\} = \{1,2,3\}$.

In other words, we obtain three valid responses $\mathrm{RSP}^{(1)}_{\theta^*,j}$, $\mathrm{RSP}^{(2)}_{\theta^*,j}$, $\mathrm{RSP}^{(3)}_{\theta^*,j}$ for all the challenges $1, 2, 3$ with respect to the same commitment CMT^*_j. Due to the computational binding property of the COM scheme, $\mathcal{C}$ is able to extract ζ^* of form

$$\zeta^* = (p^*, \mathbf{a}^*_1, \mathbf{b}^*_1, \mathbf{a}^*_2, \mathbf{b}^*_2, t^*, \mathbf{r}^*, \mathbf{v}^*, \mathbf{x}^*, g^*_1, \mathbf{e}^*_{1,1}, \mathbf{e}^*_{1,2}, g^*_2, \mathbf{e}^*_{2,1}, \mathbf{e}^*_{2,2})$$

such that $t^* \in \mathcal{T}_d$, $\mathbf{r}^*, \mathbf{v}^*$ have infinity bound β, $\mathbf{g}_1^*, \mathbf{e}_{1,1}^*, \mathbf{e}_{1,2}^*, g_2^*, \mathbf{e}_{2,1}^*, \mathbf{e}_{2,2}^*$ have infinity bound B, $\mathbf{x}^*$ has infinity bound 1; and equations $\mathbf{B} \cdot \mathbf{x}^* = p^*$ and

$$\mathbf{A}_{t^*} \cdot \mathbf{v}^* = u + \mathbf{F} \cdot \mathsf{rdec}(\mathbf{F}_0 \cdot \mathbf{r}^* + \mathbf{F}_1 \cdot \mathsf{rdec}(p^* \| \mathbf{a}_1^* \| \mathbf{b}_1^* \| \mathbf{a}_2^* \| \mathbf{b}_2^*))$$

hold, and $\mathbf{c}_1^*, \mathbf{c}_2^*$ are ciphertexts of $\mathsf{rdec}(p^*)$ under the key $(\mathbf{a}_1^*, \mathbf{b}_1^*)$ and $(\mathbf{a}_2^*, \mathbf{b}_2^*)$ with randomness $(g_1^*, \mathbf{e}_{1,1}^*, \mathbf{e}_{1,2}^*)$ and $(g_2^*, \mathbf{e}_{2,1}^*, \mathbf{e}_{2,2}^*)$, respectively.

Since $\mathcal{A}$ wins the game, then either (i) the Open algorithm outputs $\perp$ or (ii) the Open algorithm outputs $(p', \Pi_{\mathsf{open}}^*)$ with $p' \neq \perp$ but the proof Π_{open}^* is not accepted by the Judge algorithm.

By the unforgeability of the underlying signature scheme, with overwhelming probability, $(p^*, \mathbf{a}_1^*, \mathbf{b}_1^*, \mathbf{a}_2^*, \mathbf{b}_2^*, t^*, \mathbf{r}^*, \mathbf{v}^*)$ is a certificate returned by the Enroll oracle. In other words, p^* is a registered user. If p^* is a non-traceable user, then $\mathcal{A}$ does not hold the user secret key of p^*, denoted as $\mathbf{x}'$. Note that this is ensured by the definition of traceability described in Section 2.8. With probability $\geq 1/2$, $\mathbf{x}^* \neq \mathbf{x}'$ by Lemma 5, in which case we obtain a vector $\mathbf{y} = \mathbf{x}^* - \mathbf{x}' \neq \mathbf{0}$ so that $\mathbf{B} \cdot \mathbf{y} = 0$ and $\|\mathbf{y}\|_\infty \leq \|\mathbf{x}^*\|_\infty + \|\mathbf{x}'\|_\infty \leq 2$. This solves a RSIS instance. Therefore, the Open algorithm outputs $\perp$ with negligible probability. In other words, case (i) happens with negligible probability. On the other hand, if p^* is a traceable user. Then by the correctness of the underlying encryption scheme, the Open algorithm will output p^*. Furthermore, by the honest behaviour of decryption (performed by the honest challenger), the Judge algorithm always outputs 1. This implies case (ii) occurs with negligible probability. This concludes the proof.

Lemma 8. *Assuming the hardness of the* RSIS *problem, in the random oracle model, the given accountable tracing signature scheme is non-frameable.*

Proof. We show that the success probability ϵ of $\mathcal{A}$ against non-frameability is negligible assuming the hardness of solving a RSIS instance.

Let $\mathcal{C}$ be the challenger and faithfully run the experiment $\mathbf{Exp}_{\mathsf{ATS},\mathcal{A}}^{\mathsf{NF}}(\lambda)$. When $\mathcal{A}$ halts, it outputs the tuple $(M^*, \Pi_{\mathsf{gs}}^*, \mathbf{c}_1^*, \mathbf{c}_2^*, p^*, \Pi_{\mathsf{open}}^*)$. Let us consider the case that $\mathcal{A}$ wins.

The fact that $\mathcal{A}$ wins the game implies $(\Pi_{\mathsf{gs}}^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is a valid signature of the message M^* that was not obtained from queries. By the same extraction technique as in Lemma 7, we can extract witness $\mathbf{x}' \in R_q^m$ and $p' \in R_q$ such that $\|\mathbf{x}'\|_\infty \leq 1$, $\mathbf{B} \cdot \mathbf{x}' = p'$ and $\mathbf{c}_1^*, \mathbf{c}_2^*$ are correct encryptions of $\mathsf{rdec}(p')$. By the correctness of the underlying encryption scheme, $\mathbf{c}_1^*$ will be decrypted to p'.

The fact that $\mathcal{A}$ wins the game also implies Π_{open}^* passes the verification process of the Judge algorithm. Due to the soundness of the argument system that is used to generate Π_{open}^*, $\mathbf{c}_1^*$ will be decrypted to p^*. Hence we have $p' = p^*$. We observe that $\mathcal{A}$ wins the game also implies that $\mathcal{A}$ does not know the user secret key $\mathbf{x}^*$ that corresponds to p^*. Thus we obtain: $\mathbf{B} \cdot \mathbf{x}' = p' = p^* = \mathbf{B} \cdot \mathbf{x}^*$, where $\|\mathbf{x}^*\|_\infty \leq 1$. Lemma 5 implies that $\mathbf{x}' \neq \mathbf{x}^*$ with probability at least $1/2$. If they are not equal, we obtain a vector $\mathbf{y} = \mathbf{x}' - \mathbf{x}^* \neq \mathbf{0}$ such that $\mathbf{B} \cdot \mathbf{y} = 0$ and $\|\mathbf{y}\|_\infty \leq \|\mathbf{x}^*\|_\infty + \|\mathbf{x}'\|_\infty \leq 2$. However, under the hardness of the RSIS problem, the success probability of $\mathcal{A}$ is negligible. This concludes the proof.

Lemma 9. *Assuming the hardness of the* RLWE *problem, in the random oracle model, the given accountable tracing signature scheme is anonymous with accountability.*

Proof. The proof of this lemma is similar to Lemma 6 except that we do not need to switch between two decryption keys. This is because the randomized keys in the certificate of the challenged users are obtained from the pairs $(\mathbf{a}_1^{(0)}, \mathbf{b}_1^{(0)})$ and $(\mathbf{a}_2^{(0)}, \mathbf{b}_2^{(0)})$, which are not related to the opening key. The details are omitted here.

Lemma 10. *Assuming the hardness of the* RLWE *problem, in the random oracle model, the given accountable tracing signature scheme is trace-oblivious.*

Proof. We proceed through a sequence of hybrids. Let W_i be the event that adversary outputs 1 in Game i.

Game 0**:** Let this game be the experiment $\mathbf{Exp}^{\mathsf{TO}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda)$, where the adversary receives cert for user p of his choice. Parse cert as $(p, \mathbf{a}'_1, \mathbf{b}'_1, \mathbf{a}'_2, \mathbf{b}'_2, t, \mathbf{r}, \mathbf{v})$. Note that $(\mathbf{a}'_1, \mathbf{b}'_1)$ and $(\mathbf{a}'_2, \mathbf{b}'_2)$ are randomized keys from $(\mathbf{a}_1^{(0)}, \mathbf{b}_1^{(0)})$ and $(\mathbf{a}_2^{(0)}, \mathbf{b}_2^{(0)})$, respectively. We then have $\Pr[W_0] = \Pr[\mathbf{Exp}^{\mathsf{TO}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$.

Game 1**:** We modify Game 0 by replacing $(\mathbf{a}'_1, \mathbf{b}'_1)$ with a new fresh key $(\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{b}}_1)$ generated by the KeyGen algorithm of our KOE scheme. It then follows from the key randomizability of our encryption scheme, this modification is negligible to the adversary. Therefore, we have $|\Pr[W_0] - \Pr[W_1]| = \mathrm{negl}(\lambda)$.

Game 2**:** We modify Game 1 by replacing $(\mathbf{a}'_2, \mathbf{b}'_2)$ with a new fresh key $(\widetilde{\mathbf{a}}_2, \widetilde{\mathbf{b}}_2)$ as in Game 1. By the same argument, we have $|\Pr[W_1] - \Pr[W_2]| = \mathrm{negl}(\lambda)$.

Game 3**:** We change Game 2 by replacing $(\widetilde{\mathbf{a}}_2, \widetilde{\mathbf{b}}_2)$ with $(\mathbf{a}'_2, \mathbf{b}'_2)$ that are randomized key from $(\mathbf{a}_2^{(1)}, \mathbf{b}_2^{(1)})$. By the key randomizability of our encryption scheme, we have $|\Pr[W_2] - \Pr[W_3]| = \mathrm{negl}(\lambda)$.

Game 4**:** We change Game 3 by replacing $(\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{b}}_1)$ with $(\mathbf{a}'_1, \mathbf{b}'_1)$ that are randomized key from $(\mathbf{a}_1^{(1)}, \mathbf{b}_1^{(1)})$. We then have $|\Pr[W_3] - \Pr[W_4]| = \mathrm{negl}(\lambda)$. This is exactly the experiment $\mathbf{Exp}^{\mathsf{TO}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda)$. Therefore, we obtain $\Pr[W_4] = \Pr[\mathbf{Exp}^{\mathsf{TO}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]$.

Therefore, we obtain $|\Pr[\mathbf{Exp}^{\mathsf{TO}-1}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{TO}-0}_{\mathsf{ATS},\mathcal{A}}(\lambda) = 1]| = \mathrm{negl}(\lambda)$. This implies that our scheme is trace-oblivious.

Acknowledgements

The research is supported by Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S). Khoa Nguyen is also supported by the Gopalakrishnan – NTU Presidential Postdoctoral Fellowship 2018.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
2. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.
3. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.
4. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 551–572. Springer, 2014.
5. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS 2015*, volume 9326 of *LNCS*, pages 305–325. Springer, 2015.
6. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
7. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *CCS 2004*, pages 168–177. ACM, 2004.
8. J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of fully dynamic group signatures. In *ACNS 2016*, volume 9696 of *LNCS*, pages 117–136, 2016.
9. C. Boschini, J. Camenisch, and G. Neven. Floppy-sized group signatures from lattices. In *ACNS 2018*, volume 10892 of *LNCS*, pages 163–182. Springer, 2018.
10. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012*, pages 309–325. ACM, 2012.
11. E. F. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC 2000*, volume 1751 of *LNCS*, pages 276–292. Springer, 2000.
12. J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN 2012*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
13. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
14. S. Cheng, K. Nguyen, and H. Wang. Policy-based signature scheme from lattices. *Des. Codes Cryptography*, 81(1):43–74, 2016.
15. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *CRYPTO 2014*, volume 8616 of *LNCS*, pages 335–352. Springer, 2014.
16. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. *IACR Cryptology ePrint Archive*, 2014:495, 2014.
17. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
18. T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO 1984*, volume 196 of *LNCS*, pages 10–18, 1984.
19. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.

20. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 395–412. Springer, 2010.
21. A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 663–680. Springer, 2012.
22. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.
23. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 571–589. Springer, 2004.
24. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *Int. Journal of Security and Networks*, 1(1):24–45, 2006.
25. M. Kohlweiss and I. Miers. Accountable metadata-hiding escrow: A group signature case study. *PoPETs*, 2015(2):206–221, 2015.
26. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013.
27. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.
28. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*, volume 10032 of *LNCS*, pages 373–403. Springer, 2016.
29. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *ASIACRYPT 2016*, volume 10032 of *LNCS*, pages 101–131. Springer, 2016.
30. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016.
31. B. Libert, F. Mouhartem, and K. Nguyen. A lattice-based group signature scheme with message-dependent opening. In *ACNS 2016*, volume 9696 of *LNCS*, pages 137–155. Springer, 2016.
32. S. Ling, K. Nguyen, A. Roux-Langlois, and H. Wang. A lattice-based group signature scheme with verifier-local revocation. *Theor. Comput. Sci.*, 730:1–20, 2018.
33. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, 2013.
34. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC 2015*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015.
35. S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *ACNS 2017*, volume 10355 of *LNCS*, pages 293–312. Springer, 2017.
36. S. Ling, K. Nguyen, H. Wang, and Y. Xu. Constant-size group signatures from lattices. In *PKC 2018*, volume 10770 of *LNCS*, pages 58–88. Springer, 2018.
37. V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.
38. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.

39. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP 2006*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006.
40. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE 2008*, volume 5086 of *LNCS*, pages 54–72. Springer, 2008.
41. V. Lyubashevsky and G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT 2017*, volume 10210 of *LNCS*, pages 293–323. Springer, 2017.
42. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
43. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013.
44. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 204–224. Springer, 2018.
45. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.
46. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
47. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC 1990*, pages 427–437. ACM, 1990.
48. P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC 2015*, volume 9020 of *LNCS*, pages 401–426. Springer, 2015.
49. C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *STOC 2017*, pages 461–473. ACM, 2017.
50. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
51. R. D. Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. *IACR Cryptology ePrint Archive*, 2018:779, 2018. Accepted to ACM CCS 2018.
52. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.
53. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing 2012*, volume 7708 of *LNCS*, pages 270–294. Springer, 2012.
54. Y. Sakai, J. C. N. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *PKC 2012*, volume 7293 of *LNCS*, pages 715–732. Springer, 2012.
55. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS 1994*, pages 124–134. IEEE Computer Society, 1994.
56. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009.
57. J. Stern. A new paradigm for public key identification. *IEEE Trans. Information Theory*, 42(6):1757–1768, 1996.
58. K. Xagawa. Improved (hierarchical) inner-product encryption from lattices. *IACR Cryptology ePrint Archive*, 2015:249, 2015.