Accountable Tracing Signatures from Lattices

San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu(✉)

Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, 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 [12]. 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,

M. Matsui (Ed.): CT-RSA 2019, LNCS 11405, pp. 556–576, 2019.
https://doi.org/10.1007/978-3-030-12612-4_28

3,7,8,22,23,52], 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) [51], 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 [24] 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., [1–3,6]) 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 [22], 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 [24], 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 [53]. 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 [19]. Subsequently, numerous schemes offering improvements in terms of security and efficiency have been proposed [9,11,25,27,29,33,46,49]. 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 [27,30,31,34,35] 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 [30] 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 [24], 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. [35].

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 [24] 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 [17], 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., [18,50]. 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) [42], 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 [24] 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 [4,5,36,37,40,43] 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 [13,28,29,32,33,35] developed decomposition/ extension/permutation

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

techniques that operate in Stern's framework [55]. 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 [28]. More precisely, as demonstrated by Libert et al. [28] 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 [28], 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 [35], 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 [14,15] based on the hardness of the Ring Short Integer Solution (RSIS) problem). We note that the scheme from [35] 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 [24]. 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.

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 $\mathbb{Z}_q^n$. When working with vectors and matrices over R_q, we generalize the notations τ 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 $a = a_0 + a_1 \cdot X + \ldots + a_{n-1} \cdot X^{N-1} \in R$, we define $\|a\|_\infty = \max_i(|a_i|)$. Similarly, for vector $\mathbf{b} = (b_1, \ldots, b_\mathfrak{m})^\top \in R^\mathfrak{m}$, 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 ([38,39,48]). *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 [26,38,48]).

Definition 2 ([41,42,54]). *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 [10,26,41,47]). It is shown that the hardness of the RLWE problem is preserved when the secret s is sampled from the error distribution χ (see [10,41]).

2.2 Decompositions

In this work, we employ the decomposition technique from [32]. 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 $j \in [\delta_B]$. Then there is a decomposition procedure that on input $v \in [0, B]$, it outputs $\mathsf{idec}_B(a) = (a^{(1)}, a^{(2)}, \ldots, a^{(\delta_B)})^\top \in \{0,1\}^{\delta_B}$ satisfying $(B_1, B_2, \ldots, B_{\delta_B}) \cdot \mathsf{idec}_B(a) = a$.

In [35], 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 $\mathsf{rdec}_B(a) = \mathbf{a}$.

When working with vectors of ring elements, e.g., $\mathbf{v} = (v_1, \ldots, v_m)\top$ such that $\|\mathbf{v}\|_\infty \leq B$, then we let $\mathsf{rdec}_B(\mathbf{v}) = (\mathsf{rdec}_B(v_1) \| \cdots \| \mathsf{rdec}_B(v_m)) \in R^{m\delta_B}$. Now, $\forall m, B \in \mathbb{Z}^+$, we define matrices $\mathbf{H}_B \in \mathbb{Z}^{n \times n\delta_B}$ as

$$\mathbf{H}_B = \begin{bmatrix} B_1 \ldots B_{\delta_B} & & \\ & \ddots & \\ & & B_1 \ldots B_{\delta_B} \end{bmatrix}.$$

Then we have

$$\tau(a) = \mathbf{H}_B \cdot \tau(\mathsf{rdec}_B(a)) \bmod q.$$

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 [14,15], which is used to enroll new users in our construction.

Following [14,15], 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 $t = (t_0, t_1, \ldots, t_{c_j})^\top \in \mathcal{T}_j$ 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 $t_{[i]} = (t_{c_{i-1}}, \ldots, t_{c_i - 1})^\top$ 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 $c_0, \ldots, c_d$ be as above.
– Let $S \in \mathbb{Z}$ be a state that is 0 initially.

The public verification key consists of the following:

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

while the secret signing key is a Micciancio-Peikert [44] 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 $t = (t_0, t_1 \ldots, t_{c_d-1})^\top \in \mathcal{T}_d$, where $S = \sum_{j=0}^{c_d-1} 2^j \cdot t_j$, and compute $\mathbf{A}_t = [\mathbf{A}|\mathbf{A}_{[0]} + \sum_{i=1}^{d} t_{[i]}\mathbf{A}_{[i]}] \in R_q^{1\times(\overline{m}+k)}$. 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 ([14,15]). *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 Stern-Like Zero-Knowledge Argument of Knowledge

The statistical zero-knowledge arguments of knowledge (ZKAoK) presented in this work are Stern-like [55] protocols. In particular, they are Σ-protocols in the generalized sense defined in [4,20] (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. [21]. Subsequently, it was empowered by Ling et al. [32] 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. [27] put forward an abstraction of Stern's protocol to capture a wider range lattice-based relations.

2.5 Key-Oblivious Encryption

We next recall the definitions of key-oblivious encryption (KOE), as introduced in [24]. 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.

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

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

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 $\mathtt{m}$,

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

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). Details of these requirements are referred to [24] or the full version of this paper.

2.6 Accountable Tracing Signatures

An ATS scheme [24] 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 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 [24]. Specifically, we omit the time epoch when the user joins the group, since we do not consider forward and backward tracing scenarios as in [24].

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). Details of these requirements are referred to [24] or the full version of this paper.

3 Key-Oblivious Encryption from Lattices

In [24], Kohlweiss and Miers constructed a KOE scheme based on ElGamal cryptosystem [17]. 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 Sect. 4.

3.1 Description

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 Sect. 2.2).

3.2 Analysis

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 1. *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 1 is deferred to the full version of this paper.

4 Accountable Tracing Signatures from Lattices

In this section, we construct our ATS scheme based on: (i) The Ducas-Micciancio signature scheme (as recalled in Sect. 2.3); (ii) The KOE scheme described in Sect. 3; and (iii) Stern-like ZK argument systems. Due to space restriction, the details of our Stern-like ZK protocol are deferred to the full version.

4.1 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 [16]. 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 [45].

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 Sect. 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 [16].
- Choose a statistically hiding and computationally binding commitment scheme from [21], 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 Sect. 2.3.

– Initialize the Naor-Yung double-encryption mechanism [45] with the key-oblivious encryption scheme described in Sect. 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:

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

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 Sect. 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

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

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

– 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 Sect. 2.3, we have $\mathbf{r} \in R^{\overline{m}}$, $\mathbf{v} \in R^{\overline{m}+k}$ and

$$\begin{cases} \mathbf{A}_t \cdot \mathbf{v} = \mathbf{F} \cdot \mathsf{rdec}(\mathbf{F}_0 \cdot \mathbf{r} + \mathbf{F}_1 \cdot \mathsf{rdec}(\mathfrak{m})) + u, \\ \|\mathbf{r}\|_\infty \leq \beta, \quad \|\mathbf{v}\|_\infty \leq \beta. \end{cases} \tag{2}$$

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

$$\zeta = (p, \mathbf{a}'_1, \mathbf{b}'_1, \mathbf{a}'_2, \mathbf{b}'_2, t, \mathbf{r}, \mathbf{v}, \mathbf{x}, g'_1, \mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}, g'_2, \mathbf{e}'_{2,1}, \mathbf{e}'_{2,2}) \tag{3}$$

such that

(i) The conditions in (2) 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 our Stern-like ZK protocol. The protocol is repeated $\kappa = \omega(\log \lambda)$ times and made non-interactive via Fiat-Shamir heuristic [16] 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

$$\xi = (\mathbf{A}, \mathbf{A}_{[0]}, \ldots, \mathbf{A}_{[d]}, \mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, u, \mathbf{B}, \mathbf{c}_1, \mathbf{c}_2) \tag{4}$$

– 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 (4).

– For each $i \in [\kappa]$, run the verification phase of our Stern-like ZK protocol 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 Sect. 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.

$$\begin{cases} \|s_1\|_\infty \leq B;\ \|\mathbf{e}_1\|_\infty \leq B;\ \|\mathbf{y}\|_\infty \leq \lceil q/10 \rceil; \\ \mathbf{a}_1^{(1)} \cdot s_1 + \mathbf{e}_1 = \mathbf{b}_1^{(1)}; \\ \mathbf{c}_{1,2} - \mathbf{c}_{1,1} \cdot s_1 = \mathbf{y} + \lfloor q/4 \rfloor \cdot \mathsf{rdec}(p'). \end{cases} \tag{5}$$

Since the conditions in (5) only encounter linear secret objects with bounded norm, we can easily handled them using the Stern-like techniques. 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

$$\mathrm{CH} = \mathcal{H}_{\mathsf{FS}}\big(\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, \mathbf{a}_1^{(1)}, \mathbf{b}_1^{(1)}, M, \Sigma, p'\big). \tag{6}$$

– 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 (2). 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 (1) hold. If not, output 0.
– Else output 1.

4.2 Analysis of Our ATS Scheme

Efficiency. We first analyze the efficiency of our scheme from Sect. 4.1 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 $\mathcal{O}(\lambda^2 \cdot \log^3 \lambda) \cdot \omega(\log \lambda) = \widetilde{\mathcal{O}}(\lambda^2)$.
– 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 (3). 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 Sect. 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 Sect. 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 2, we prove that our scheme satisfies the security requirements of accountable tracing signatures, as specified by Kohlweiss and Miers.

Theorem 2. *Under the* RLWE *and* RSIS *assumptions, the accountable tracing signature scheme described in Sect. 4.1 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.*

The proof of Theorem 2 is deferred to the full version of this paper.

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. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_16
2. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38
3. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: the case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_11
4. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 551–572. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_29
5. Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9326, pp. 305–325. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24174-6_16
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_3
7. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: CCS 2004, pp. 168–177. ACM (2004)
8. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 117–136. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_7

9. Boschini, C., Camenisch, J., Neven, G.: Floppy-sized group signatures from lattices. In: Preneel, B., Vercauteren, F. (eds.) ACNS 2018. LNCS, vol. 10892, pp. 163–182. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93387-0_9
10. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: ITCS 2012, pp. 309–325. ACM (2012)
11. Camenisch, J., Neven, G., Rückert, M.: Fully anonymous attribute tokens from lattices. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 57–75. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_4
12. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
13. Cheng, S., Nguyen, K., Wang, H.: Policy-based signature scheme from lattices. Des. Codes Cryptogr. **81**(1), 43–74 (2016)
14. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 335–352. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_19
15. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. IACR Cryptology ePrint Archive 2014, 495 (2014)
16. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
17. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_2
18. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206. ACM (2008)
19. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_23
20. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 663–680. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_40
21. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_23
22. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_34
23. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. Int. J. Secur. Netw. **1**(1), 24–45 (2006)
24. Kohlweiss, M., Miers, I.: Accountable metadata-hiding escrow: a group signature case study. PoPETs **2015**(2), 206–221 (2015)
25. Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-based group signatures with logarithmic signature size. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 41–61. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_3
26. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. **75**(3), 565–599 (2015)

27. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 373–403. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_13
28. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 101–131. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_4
29. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 1–31. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_1
30. Libert, B., Mouhartem, F., Nguyen, K.: A lattice-based group signature scheme with message-dependent opening. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 137–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_8
31. Ling, S., Nguyen, K., Roux-Langlois, A., Wang, H.: A lattice-based group signature scheme with verifier-local revocation. Theor. Comput. Sci. **730**, 1–20 (2018)
32. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 107–124. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_8
33. Ling, S., Nguyen, K., Wang, H.: Group signatures from lattices: simpler, tighter, shorter, ring-based. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 427–449. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_19
34. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-based group signatures: achieving full dynamicity with ease. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 2017. LNCS, vol. 10355, pp. 293–312. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61204-1_15
35. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Constant-size group signatures from lattices. In: Abdalla, M., Dahab, R. (eds.) PKC 2018. LNCS, vol. 10770, pp. 58–88. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76581-5_3
36. Lyubashevsky, V.: Fiat-Shamir with aborts: applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_35
37. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_43
38. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_13
39. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: a modest proposal for FFT hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-71039-4_4
40. Lyubashevsky, V., Neven, G.: One-shot verifiable encryption from lattices. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 293–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_11

41. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
42. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. J. ACM **60**(6), 43:1–43:35 (2013)
43. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 204–224. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_8
44. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
45. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437. ACM (1990)
46. Nguyen, P.Q., Zhang, J., Zhang, Z.: Simpler efficient group signatures from lattices. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 401–426. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_18
47. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: STOC 2017, pp. 461–473. ACM (2017)
48. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_8
49. del Pino, R., Lyubashevsky, V., Seiler, G.: Lattice-based group signatures and zero-knowledge proofs of automorphism stability. IACR Cryptology ePrint Archive, 2018:779 (2018). Accepted to ACM CCS 2018
50. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93. ACM (2005)
51. Sakai, Y., Emura, K., Hanaoka, G., Kawai, Y., Matsuda, T., Omote, K.: Group signatures with message-dependent opening. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 270–294. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36334-4_18
52. Sakai, Y., Schuldt, J.C.N., Emura, K., Hanaoka, G., Ohta, K.: On the security of dynamic group signatures: preventing signature hijacking. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 715–732. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_42
53. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: FOCS 1994, pp. 124–134. IEEE Computer Society (1994)
54. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_36
55. Stern, J.: A new paradigm for public key identification. IEEE Trans. Inf. Theory **42**(6), 1757–1768 (1996)