Lattice-Based Group Signatures in the Standard Model, Revisited

Nam Tran[1,2⋆], Khoa Nguyen[1], Dongxi Liu[2], Josef Pieprzyk[2,3], and Willy Susilo[1] ⋆⋆

[1] University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia
ndt141@uowmail.edu.au, khoa@uow.edu.au, wsusilo@uow.edu.au
[2] CSIRO Data61, Marsfield, 26 Pembroke Road, NSW 2122, Australia
[3] Institute of Computer Science, Polish Academy of Sciences, Poland
Dongxi.Liu@data61.csiro.au, Josef.Pieprzyk@data61.csiro.au

Abstract. The study of lattice-based group signatures has been a prominent research direction since 2010. While recent advances in the field have yielded schemes in the random oracle model with strong security properties and nearly practical efficiency, the current state of affairs for lattice-based group signatures in the standard model is still much less satisfactory. Existing schemes, proposed by Katsumata and Yamada (EUROCRYPT'19) or implied by generic non-interactive zero-knowledge proofs for NP (by Peikert and Shiehian at CRYPTO'19 and by Waters at STOC'24), either only fulfil a weak notion of anonymity called selfless anonymity, or require a strong lattice assumption, or suffer from extremely large signatures and/or public keys.

This work aims to enhance the state of affairs for lattice-based group signatures in the standard model. We provide improved constructions that simultaneously achieves: (i) signature and public key sizes significantly smaller than those of known schemes; (ii) full anonymity in the CPA and CCA senses; (iii) security based on standard SIS and LWE assumptions with polynomial approximation factors regarding worst-case lattice problems (in general lattices). Our design approach slightly departs from that of existing pairing-based and lattice-based constructions. In the design process, we adapt and develop several lattice-based cryptographic ingredients that may be of independent interest. At the heart of our constructions is a reasonably efficient non-interactive zero-knowledge proof system for relations typically appearing in advanced privacy-preserving lattice-based cryptographic protocols. These relations are addressed by a trapdoor Σ-protocol with an inverse polynomial soundness error, which is made non-interactive via the standard-model Fiat-Shamir transform of Canetti et al. (STOC'19) and a compiler by Libert et al. (ASIACRYPT'20).

⋆ Nam Tran is supported by CSIRO Data61 PhD Scholarship and CSIRO Data61 Top-up Scholarship.

⋆⋆ Willy Susilo is supported by the Australian Research Council Australian Laureate Fellowship FL230100033.

1 Introduction

Group signature, introduced in the seminal work of Chaum and van Heyst [23], is a fundamental privacy-preserving primitive allowing registered users of a group to anonymously sign messages on behalf of the whole group. Yet, users are kept accountable for their actions since the signer of any problematic signature can be traced by an opening authority – should the need arise. These two appealing features enable group signatures to find applications in various real-life scenarios, such as private-and-accountable access to public transport, anonymous online communications, e-bidding and e-commerce systems, just to name a few. On the theoretical front, designing secure and efficient group signature schemes is interesting and challenging, since those advanced constructions usually require a sophisticated combination of several cryptographic building blocks. The present work focuses on group signatures from lattices.

Background and Motivation. The study of lattice-based group signatures, initiated by Gordon et al. [34], has been a prominent and highly active research direction since the rise of lattice-based cryptography [65,31,30,57]. There have been proposed various improvements in terms of security and efficiency. In the following, we will give a brief overview on the development progress of lattice-based group signatures, categorized into schemes proven secure in the random oracle model (ROM) and those in the standard model.

Lattice-Based Group Signatures in the ROM. The pioneering work by Gordon et al. [34] yielded a scheme with a signature size linear in the number of group users N. While the scheme only satisfies CPA-full-anonymity [8], in which the anonymity adversary is not given access to the opening oracle, the strongest notion of CCA-full-anonymity [5] was then achieved in a follow-up work by Camenisch et al. [18]. Then the linear barrier was overcome by Laguillaumie et al. [41], who put forward the first scheme with a signature size logarithmic in N, at the cost of relatively large parameters. Simpler and more efficient solutions with $\mathcal{O}(\log N)$ signature size and some additional functionalities were then proposed in subsequent works [42,50,63,44,51,69]. Ling et al. [52] kept up the improvement in terms of asymptotic efficiency by introducing a scheme with constant-size signatures, i.e., $\mathcal{O}(1)$ in N.

However, all these schemes are not concretely efficient, due to a large dependency on other major parameters, such as security parameter λ and lattice dimension n. A recent line of work [26,11,56,55,6,10] has significantly improved the concrete efficiency of lattice-based group signatures, producing signature sizes of just about tens of KBs. It is worth noting that all the aforementioned schemes achieve in the ROM the standard notions of CPA/CCA-full-anonymity, with one exception: the scheme from [42] only satisfies a weaker notion of anonymity, namely, selfless-anonymity [15,9]. While full-anonymity allows the adversary to corrupt all group users, selfless-anonymity prevents it from exposing the signing keys of the two users in question. In other words, signatures do not remain anonymous to anyone knowing the signing keys (and in particular, to the signers themselves). This contrast is easily seen in the borderline case where the group

has exactly two users: while full-anonymity allows the adversary to obtain the signing keys of both users in the group, selfless-anonymity does not allow the adversary to know any of the keys.

Lattice-Based Group Signatures in the Standard Model. Achieving security in the standard model was a particularly challenging obstacle in the development of lattice-based group signatures. The main reason was the then lack of lattice-based non-interactive zero-knowledge (NIZK) proofs in the standard model. A breakthrough was made by Katsumata and Yamada [37] who introduced a novel approach for obtaining group signatures without NIZKs, based on a special type of attribute-based signatures (ABS) [58] (which can be viewed as a kind of designated-prover NIZKs [40]) and a secret-key encryption (SKE) scheme with some additional properties. Their results yielded several interesting constructions of lattice-based group signatures in the standard model. Their first instantiation (henceforth, **KY19-I**) is a scheme relying on standard Short-Integer-Solution (SIS) [2] and Learning-With-Errors (LWE) [65] assumptions with polynomial approximation factors (w.r.t. the underlying worst-case lattice problems). However, its signature size and public key size are linear in N. Their second construction (henceforth, **KY19-II**) achieves constant-size signature and public key, but it has to rely on a relatively strong assumption: subexponential hardness of SIS. It is worth noting that, for both constructions, the dependencies on security parameter λ and lattice dimension n are quite heavy (see Table 1 for more details). Furthermore, in terms of anonymity property, these constructions only achieve selfless-anonymity (in the CCA sense). A high-level reason for this limitation is that the KY19 design approach does not make use of public-key encryption – which has been shown [1] to be necessary for achieving full-anonymity.

Shortly after the publication of [37], Peikert and Shiehian [64] solved the long-standing problem of constructing NIZKs for all NP statements based on the standard LWE assumption in the standard model, using techniques inspired by Canetti *et al.* [20,19], which make use of *correlation intractable* (CI) hash functions. Recently, another line of work [67,68,13] successfully realized a primitive called *hidden bit generator* (HBG) from lattice assumptions and as a corollary they obtain a generic NIZK based on the NIZK system in the hidden bit model of Feige, Lapidot and Shamir [28]. Theoretically speaking, these results imply the feasibility of designing lattice-based group signatures achieving full-anonymity in the standard model. However, to apply their protocol, one would need to convert statements of the form "I have a valid membership certificate from the group manager, and I have honestly encrypted my identity" typically used in building group signatures following the sign-then-encrypt-then-prove paradigm to an instance of the Graph Hamiltonicity problem. Such a conversion would require an expensive Karp reduction and would lead to a significantly large group signature size, estimated of order $\Omega(n^6\lambda\log^3\lambda)$. Achieving noticeably better efficiency for NIZK-based group signatures from lattices in the standard model has been a tough open question for the last 5 years.

As a summary, while practically relevant lattice-based group signatures with strong security properties have been obtained in the ROM, the situation for

their standard-model counterparts is much less satisfactory: known schemes either only fulfil the weak notion of selfless-anonymity, or require a strong lattice assumption, or suffer from extremely large signatures and/or public keys. This unpleasant state of affairs inspires us to investigate the problem of

Constructing lattice-based group signatures in the standard model simultaneously featuring reasonably short signatures and keys, relying on mild lattice assumptions and achieving full anonymity.

We remark that we do not aim to achieve practical efficiency for lattice-based group signatures in the standard model, given the current lack of truly practical NIZKs without the ROM. For advanced cryptographic primitives, there is usually a big research gap between theoretical feasibility and concrete efficiency, which may require many steps of development to fill in. We can observe such situations in primitives like MPC and FHE, and, in particular, in the development of lattice-based group signatures in the ROM. As we discussed above, during 2010 – 2018, the focus was to improve asymptotic efficiency, from signature size $\mathcal{O}(\lambda^2 \cdot N)$ in [34,18] to $\mathcal{O}(\lambda \cdot \log N)$ in [41,50,63,44], and then to constant-size signature $\mathcal{O}(\lambda)$ in [52]. Once the achieved asymptotic efficiency had been somewhat optimal, we then have witnessed the introduction of schemes with concrete efficiency such as [26,11,56,55,6,10] since 2018. Hence, for lattice-based group signatures in the standard model, we believe it is important at this point to improve the asymptotic efficiency and reduce the currently large gap between theory and practice. Even though such attempts may not directly yield a practically-interesting construction, they may inspire follow-up works that will eventually lead to concretely efficient schemes.

Our Contributions. In this work, we provide an affirmative answer to the problem discussed above. Specifically, we put forward a lattice-based group signature scheme in the standard model that enjoy the following properties:

- Signature size $\mathcal{O}(n\lambda \log^2 \lambda)$, which is significantly short signature sizes, compared to the schemes **KY19-I** and **KY19-II** from [37] and (potential) constructions relying on generic NIZKs for NP. For the latter, we consider the NIZK based on CI hash function from [64] (that we name **NIZK-CI**), and recent constructions from hidden-bit generator [67,68,13] (that we name **NIZK-HBG**).
- Public key size $\mathcal{O}(n^2 \log^3 \lambda)$, which is smaller than those of **KY19-I** and **KY19-II** by a factor at least $\mathcal{O}(n^2 \log \lambda)$.
- Our scheme achieves CCA-full-anonymity, and relies on standard SIS and LWE assumptions with polynomial approximation factors, which is comparable to **KY19-I** and constructions based on generic NIZKs.

We summarize the comparison between our scheme and other (known and potential) lattice-based group signatures in the standard model in **Table** 1. One can observe that our construction not only enjoys strong security properties but also considerably improves the previous designs in terms of asymptotic efficiency. When it comes to concrete parameters, we estimate that for 128 bits of security and for a group of $N = 2^{20}$ users, our scheme produces signature sizes at least

Scheme	Signature	Public key	Anonymity	Assumptions
KY19-I [37]	$\mathcal{O}(Nn\log^2\lambda + n^2\log^3\lambda)$	$\mathcal{O}(Nn^4\log^4\lambda)$	CCA-Selfless	SIS LWE
KY19-II [37]	$\mathcal{O}(n^3\log^4\lambda)$	$\mathcal{O}(n^4\log^4\lambda)$	CCA-Selfless	subexp-SIS LWE
NIZK-CI [64] (via [36,31,45])	$\mathcal{O}(n^6\lambda\log^3\lambda)$	$\mathcal{O}(n^2\log^3\lambda)$	CCA-Full	SIS LWE
NIZK-HBG [67,68,13] (via [36,31,45])	$\mathcal{O}(n^6\lambda\log^3\lambda)$	$\mathcal{O}(n^2\log^3\lambda)$	CCA-Full	SIS LWE
Ours	$\mathcal{O}(n\lambda\log^2\lambda)$	$\mathcal{O}(n^2\log^3\lambda)$	CCA-Full	SIS LWE

Table 1. Comparison of signature sizes, public key sizes, anonymity notions and lattice assumptions between our scheme and other (known and potential) lattice-based group signatures in the standard model. Three governing parameters are used for the size estimations: security parameter λ, the number of group users $N \in \mathsf{poly}(\lambda)$, and lattice dimension $n = \Omega(\lambda)$. The size estimations for **KY19-I** and **KY19-II** are done based on the constructions and parameters provided in [37]. While the schemes **NIZK-CI** and **NIZK-HBG** have not been explicitly proposed, we may obtain it via the same supporting techniques as for our scheme: the signature scheme from [36], the dual-Regev encryption [31] and the compiler from [45]. Our estimation shows that these schemes have huge signature sizes, due to the expensive overhead of Karp reduction for transforming the defining relations of our scheme to a graph-based relation.

several orders of magnitude smaller than those of **KY19-I**, **KY19-II** and of the potential constructions via **NIZK-CI** and **NIZK-HBG**. The public key size of our scheme is also about two orders of magnitude smaller than those of **KY19-I** and **KY19-II**.

Technical Overview. At a high level, our construction employs the traditional sign-then-encrypt paradigm for designing fully anonymous group signatures following the BMW model [5], yet we introduce several modifications in the process. The generic approach makes use of three major building blocks. The first ingredient is an ordinary signature scheme compatible with zero-knowledge proofs of a message-signature pair, which is used by the group manager to generate membership certificates for all group members. The second ingredient is a public-key encryption scheme that will be used to encrypt the signer's identity. And the final ingredient is an NIZK proof system that can handle relations involving messages-signatures and ciphertexts-plaintexts in the aforementioned components. For the signature layer, we choose the recently proposed scheme from [36] which we refer to as JRS signature scheme. The JRS signature features a short public key and that can be based on a standard SIS assumption over general lattices. For the encryption layer, we choose the dual-Regev encryption scheme [31], which also offers a reasonably small public key and is based on the LWE assumption.

In our construction, the group public key consists of $(\mathbf{A}, \mathbf{B} = \mathbf{A}\cdot\mathbf{T}, \mathbf{C}, \mathbf{D}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \ell_{\mathbf{m}}} \times \mathbb{Z}_q^n$, where $\mathbf{T} \in \mathbb{Z}^{m\times m}$ is a small norm matrix. Unlike the generic construction from [5] where group manager signs users' pub-

lic keys, we implement a delegation mechanism, in which each user is given a trapdoor $\mathbf{R}_{\mathsf{id}}$ for a matrix that "encodes" the identity id. When issuing a group signature on a message $\mathbf{m}$, such a trapdoor allows the user to issue a JRS signature for $\mathbf{m}$. This is done using an idea similar to the one employed in [26,56,55], in which a user id is uniquely associated with a matrix

$$\mathbf{A}_{\mathsf{id}} = (\mathbf{A} \mid \mathsf{id} \cdot \mathbf{G} - \mathbf{B}),$$

where $\mathbf{G} \in \mathbb{Z}^{n \times m}$ is a gadget matrix. Via the Gaussian sampling technique of [59], the setup authority can generate the secret key of an identity $\mathsf{id} \neq 0$, which is a small norm matrix $\mathbf{R}_{\mathsf{id}} \in \mathbb{Z}^{2m \times m}$ satisfying $\mathbf{A}_{\mathsf{id}}\mathbf{R}_{\mathsf{id}} = \mathbf{C}$. The secret $\mathbf{R}_{\mathsf{id}}$ will act as the secret signing key of user id under JRS signature scheme.

We also depart from the usual signing methods adopted in previous standard-model group signature schemes [49,48] from pairings, where signers sign a message using a one-time signature and use their secret signing keys to certify the corresponding one-time verification key. Such an approach, although convenient, usually results in a blow-up of public-key size as the verification key of (lattice-based) one-time signatures is often not compact. Hence, we let signers sign directly on messages. Namely, if the message is $\mathbf{m} \in \{0,1\}^{\ell_{\mathbf{m}}}$, a user id signs $\mathbf{m}$ by generating a JRS signature $(\tau, \mathbf{x})$ consisting of a non-zero scalar tag τ and a discrete Gaussian vector $\mathbf{x} \in \mathbb{Z}^{3m}$ that satisfy

$$(\mathbf{A}_{\mathsf{id}} \mid \tau \cdot \mathbf{G} - \mathbf{C})\,\mathbf{x} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} \bmod q.$$

The above signing process differs from the original construction of [36], in which the signer also produces a short randomness $\mathbf{r}$ such that

$$(\mathbf{A}_{\mathsf{id}} \mid \tau \cdot \mathbf{G} - \mathbf{C})\,\mathbf{x} = \mathbf{u} + \mathbf{A} \cdot \mathbf{r} + \mathbf{D} \cdot \mathbf{m} \bmod q.$$

Simply put, signer creates a KTX commitment [38] on the message $\mathbf{m}$. In our construction, there is no need for committing to $\mathbf{m}$ and thus the commitment part can be ignored. With this modification, we obtain a variant of JRS signature scheme that can be viewed as a plain lattice-based counterpart of Ducas-Micciancio ring-based construction [27].

At this point, the standard approach of [5] requires encrypting id and the signature $(\tau, \mathbf{x})$, then proving in ZK that encryption is correct and $(\tau, \mathbf{x})$ is a valid JRS signature. However, encrypting the entire signature $(\tau, \mathbf{x})$ might be an overkill as we observe that certain part of $\mathbf{x}$ can be revealed while essentially leaking no information about signer's identity id. This comes from a nice property of Gaussian sampling algorithms [21,22], exploiting the fact that a discrete Gaussian vector $\mathbf{s}$ of high dimension, conditioned on a linear equation $\mathbf{U}\mathbf{s} = \mathbf{v} \bmod q$, can be sampled by sampling some components of $\mathbf{s}$ independently of the matrix part in $\mathbf{U}$. This observation serves as a key technical point of the construction from [63], and the same can be adapted in our scheme. Observe that $(\tau, \mathbf{x})$ should satisfy

$$(\mathbf{A} \mid \mathsf{id} \cdot \mathbf{G} - \mathbf{B} \mid \tau \cdot \mathbf{G} - \mathbf{C}) \cdot \mathbf{x} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} \bmod q,$$

then we can let a signer id reveal the component $\mathbf{s}_1$ and $\mathbf{s}_2$ in $\mathbf{x}$ corresponding to the matrix part $\mathsf{id} \cdot \mathbf{G} - \mathbf{B}$ and $\tau \cdot \mathbf{G} - \mathbf{C}$, but keep secret: the part $\mathbf{s}$ corresponding

to $\mathbf{A}$, the identity id and the tag τ which depends on signer's state. Conveniently, the relation showing validity of a JRS signature now becomes

$$\mathbf{A} \cdot \mathbf{s} + \mathsf{id} \cdot \mathbf{G}\mathbf{s}_1 + \tau \cdot \mathbf{G}\mathbf{s}_2 = \mathbf{u} + \mathbf{D}\mathbf{m} + \mathbf{B}\mathbf{s}_1 + \mathbf{C}\mathbf{s}_2 \bmod q,$$

which is *linear* in the secret $\mathbf{s}, \mathsf{id}$ and τ. Thus, it suffices for signer id to encrypt $\mathbf{s}, \mathsf{id}$ and τ under dual-Regev encryption, then prove well-formedness of the ciphertext as well as the validity of a JRS signature, which is a linear constraint on $\mathbf{s}, \mathsf{id}$ and τ. In particular, our method allows us to avoid *quadratic* relations, which were required in previous constructions such as [42,50,44,43,52].

It remains the task of designing a standard-model NIZK system for the signer to prove knowledge of a valid witness satisfying some linear relations, and the witness is encrypted to a given ciphertext. We therefore would need a reasonably efficient NIZK system for handling such equations. By "reasonably efficient", we mean to avoid the need to repeat an atomic zero-knowledge protocol $\Omega(\lambda)$ times to achieve a negligible soundness error, as in [45]. Our wishful thinking is that the NIZK can be done in just "one shot", i.e., the number of protocol repetitions is simply 1. While this feat has been achieved in number-theoretic-based NIZKs in the standard model (e.g., [47]) and lattice-based NIZKs in the ROM (e.g., [10]), it still remains a longstanding open problem to do so from lattices and in the standard model. Instead, we employ an atomic protocol with an inverse-polynomial soundness error $1/C$, for some $C \in \mathsf{poly}(\lambda)$. Asymptotically, the protocol needs to be repeated $O(\lambda/\log \lambda)$ times for a soundness level $1-2^{-\lambda}$. However, when it comes to concrete parameters, to achieve soundness $1-2^{-128}$, one only needs $\kappa = 8$ repetitions when $C = 2^{16}$, which is still much better than $\kappa = 128$ – if the underlying protocol has soundness error $1/2$.

Let us now discuss the techniques in more detail by considering a CPA-secure variant of our scheme. In essence, we need an NIZK for a relation specified by

$$\begin{aligned} &\mathbf{t} = \mathbf{R} \cdot \mathbf{x} \bmod q, \\ &\mathbf{c} = \begin{pmatrix} \mathbf{U}^\top \\ \mathbf{V}^\top \end{pmatrix} \mathbf{r} + \mathbf{e} + \begin{pmatrix} \mathbf{0} \\ \lfloor q/2K \rfloor \mathbf{x} \end{pmatrix} \bmod q, \end{aligned} \tag{1}$$

where the witness $\mathbf{x}$ is a small-norm vector. Essentially, the relation captures the fact that $\mathbf{x}$ satisfies some linear relation and that a dual-Regev ciphertext $\mathbf{c}$ encrypts $\mathbf{x}$. Since the relation is linear in the witness and with some constraints over the norms, this suffices for the blueprint Σ-protocol of Lyubashevsky [53,54]. In this Σ-protocol, the prover produces a response $(\mathbf{z}_\mathbf{x}, \mathbf{z}_\mathbf{r}, \mathbf{z}_\mathbf{e})$ satisfying

$$\begin{aligned} &\mathbf{a} + c \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}_\mathbf{x} \bmod q, \\ &\mathbf{v} + c \cdot \mathbf{c} = \begin{pmatrix} \mathbf{U}^\top \\ \mathbf{V}^\top \end{pmatrix} \cdot \mathbf{z}_\mathbf{r} + \mathbf{z}_\mathbf{e} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}_\mathbf{x} \end{pmatrix} \bmod q; \end{aligned} \tag{2}$$

where c is a verifier's challenge chosen from some set of small integer, and $(\mathbf{a}, \mathbf{v})$ is a prover's first message. Here, rather than using single-bit challenge as in [53], we let challenge space be the set of integer $\{0, 1, \ldots, C\}$ where C is polynomial

in the security parameter λ, thus requiring about $\lambda/\log\lambda$ executions in parallel to make the soundness error negligible.

To remove verifier's randomness and obtain an NIZK proof, one could apply the well-known Fiat-Shamir transformation [29], with the caveat that soundness can only be argued in the ROM and might not be preserved when employing concrete hash functions. Quite recently, the breakthrough result of Canetti *et al.* [19] showed that the Fiat-Shamir transformation [29] can be soundly instantiated without random oracles, assuming the underlying Σ-protocol is *trapdoor*. In simple terms, this requires that for any false statement, there is *at most one* verifier's challenge that the prover can cheat[4], and there is a *trapdoor information* (which could be language-dependent) allowing one to efficiently identify this "bad" challenge from any false statement and any prover's first message. If we consider the Σ-protocol of [53,54] that proves (1), this requires that for a false statement $(\mathbf{t}, \mathbf{c})$, we can find the unique challenge c such that there is a response $(\mathbf{z_x}, \mathbf{z_r}, \mathbf{z_e})$ satisfying verification equations of (2). A closer look shows that the second equation of (2)

$$\mathbf{v} + c \cdot \mathbf{c} = \begin{pmatrix} \mathbf{U}^\top \\ \mathbf{V}^\top \end{pmatrix} \cdot \mathbf{z_r} + \mathbf{z_e} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z_x} \end{pmatrix} \bmod q,$$

can be interpreted as "there exist encryption randomness $(\mathbf{z_r}, \mathbf{z_e})$ and a plaintext $\mathbf{z_x}$ such that $\mathbf{v} + c \cdot \mathbf{c} \bmod q$ is a dual-Regev ciphertext encrypting $\mathbf{z_x}$ with randomness $(\mathbf{z_r}, \mathbf{z_e})$". Hence, $(\mathbf{z_x}, \mathbf{z_r}, \mathbf{z_e})$ can be recovered from $\mathbf{v}+c\cdot\mathbf{c} \bmod q$, as one can embed an LWE-inversion trapdoor (e.g. Micciancio-Peikert trapdoor [59]) allowing to recover the randomness of encryption (and thus the corresponding plaintext). Such a capability enables an efficient "bad" challenge finding algorithm in a very strong sense: not only the algorithm finds the unique challenge that a dishonest prover can cheat, but it also retrieves the corresponding response the prover must produce.

Therefore, we obtain a trapdoor Σ-protocol for the relation defined by (1), by following Lyubashevksy's framework and utilizing the randomness recovery property underlying dual-Regev encryption scheme. The corresponding NIZK argument system for (1) can be achieved via Fiat-Shamir transform with correlation intractable hash functions of [19,64]. However, the resulting NIZK can only achieve a weak-form of programmability as proven in [24], and might not be enough for anonymity of group users. In particular, even for CPA-anonymity, the zero-knowledge property of the NIZK argument obtained in the framework of [19,64,24] is not sufficient for our construction, as it requires the ability to modify the key of the correlation intractable hash function, which is chosen at the time the group is set up.

To tackle the above issue, we turn to the stronger compilers of [45], which can build a multi-theorem zero-knowledge and sound NIZK system generically from *any* trapdoor Σ-protocol. In particular, they also provided a compiler turning any trapdoor Σ-protocol into a *simulation-sound* NIZK under standard lattice

[4] This is also known as *optimal soundness* [25].

assumptions. Looking ahead, our CCA-anonymous group signature employs the latter along with the standard double encryption technique from [60].

Regarding the traceability of the group signatures, we note that since the construction does not explicitly follow the generic BMW framework [5], traceability is not directly reduced to the security of the employed digital signature scheme. Furthermore, due to soundness slack of the NIZK system for relation (1), an NIZK proof only guarantees that there exists $(\overline{\mathbf{x}}, \overline{c})$ such that $\mathbf{R} \cdot \overline{\mathbf{x}} = \overline{c} \cdot \mathbf{t}$, and that $\overline{c} \cdot \mathbf{c}$ has the form of a dual-Regev ciphertext. As a consequence, a ciphertext $\mathbf{c}$ in the signature might not be well-formed and not decrypted to any identity at all. As such, to argue traceability we directly reduce the adversary to one which solves an instance of SIS problem.

Discussions and Open Questions. Our results significantly narrow the efficiency gap between group signatures in the standard model and those in the ROM. However, there is still room for improvement. For instance, one could further enhance the efficiency of the protocol in Section 3.2 by utilizing batching techniques such as in [54], or working with structured lattices to extend the size of the challenge space. Nevertheless, to the best of our knowledge these techniques are not compatible with the trapdoor Σ-protocol framework. Therefore, constructing a *one-shot* trapdoor Σ-protocol for lattice-based relations would be an interesting direction for future research. Such a protocol, if it existed, would immediately yield a group signature with efficiency rivaling that of its counterparts in the ROM, in a way similar to the ring signature of [47].

Quite recently, there has been an interesting line of work generalizing group signatures to privacy-preserving signature systems with advanced tracing capabilities [46,61,62]. In this broader context, we hope that this work may motivate further research into lattice-based constructions of these advanced primitives in the standard model.

At the heart of our group signatures is a standard model NIZK proof of a valid signature on a message for the JRS signature scheme. Upon closer inspection, such an NIZK can be extended to prove knowledge of a valid message-signature pair. Looking this way, the techniques presented in this paper may have potential applications in anonymous credentials [17,4], anonymous attestations [14], decentralized e-cash systems [16], group encryptions [39], i.e., primitives relying on NIZK proving possession of a message-signature pair as a building block. We note that constructing any of these schemes from lattices in the standard model remains an open question.

Organization. The rest of the paper is organized as follows. Section 2 recalls some definitions and results frequently used in the paper. Section 3 describes our trapdoor Σ-protocol, which serves as a core building block for the CCA-secure group signature in Section 4. Several supporting materials are provided in the Appendix.

Changelog. This version was updated in November 2026, which fixed the opening algorithms and the security proofs. We thank Minh Pham for pointing out a bug in the previous version.

2 Preliminaries

2.1 Basic Notations

Throughout this paper, vectors are treated as column vectors and denoted by bold, lower-case letters. Matrices are denoted by bold, upper-case letters. The coordinates of a vector are indexed in an array-like manner, starting from 1; for example, given an n-th dimensional vector $\mathbf{v}$, then $\mathbf{v} = (\mathbf{v}[1], \ldots, \mathbf{v}[n])$. For two vectors $\mathbf{v}_1, \mathbf{v}_2$; the concatenation is a column vector and denoted by $(\mathbf{v}_1, \mathbf{v}_2)$.

For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$. We let $\mathsf{bin}(n) \in \{0,1\}^{\lceil \log n \rceil}$ be the binary representation of n in little-endian, which satisfies $n = \mathsf{bin}(n)[1] + \mathsf{bin}(n)[2] \cdot 2 + \ldots + \mathsf{bin}(n)[\lceil \log n \rceil] \cdot 2^{\lceil \log n \rceil - 1}$.

The ℓ_2-norm and ℓ_∞-norm of $\mathbf{x} \in \mathbb{R}^n$ are respectively denoted by $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_\infty$. For a matrix $\mathbf{R} = (\mathbf{r}_1 | \cdots | \mathbf{r}_m) \in \mathbb{R}^{n \times m}$, we let $\|\mathbf{R}\|_2$ be the operator norm of the linear function defined by $\mathbf{R}$. We have that $\|\mathbf{R}\|_2 \leq \sqrt{m} \cdot \max_i \|\mathbf{r}_i\|_2$, and $\|\mathbf{R}\mathbf{x}\|_2 \leq \|\mathbf{R}\|_2 \|\mathbf{x}\|_2 \leq \sqrt{m} \cdot \max_i \|\mathbf{r}_i\|_2 \cdot \|\mathbf{x}\|_2$ for any $\mathbf{x} \in \mathbb{R}^m$. Moreover, by the Cauchy–Schwarz inequality, we have that $\|\mathbf{R}\mathbf{x}\|_\infty \leq \max_{j \in [n]} \|\mathbf{R}[j]\|_2 \cdot \|\mathbf{x}\|_2$ for any $\mathbf{x} \in \mathbb{R}^m$, where $\mathbf{R}[j]$ denotes the j-th row of $\mathbf{R}$.

For a finite set S, we let $\mathcal{U}(S)$ denote the uniform distribution over S. We write $x \leftarrow D$ when x is sampled from a probability distribution D.

2.2 Lattice-Based Cryptography

Hardness Assumptions We recall the SIS and LWE problems.

Definition 2.1. *The SIS problem* $\mathsf{SIS}_{q,n,m,B}$*, where* n, m *are dimensions,* q *is a modulus and* B *is a norm bound such that* $n < m$ *and* $0 < B < q$*, asks to find* $\mathbf{z} \in \mathbb{Z}_q^m$ *satisfying* $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q$ *and* $0 < \|\mathbf{z}\|_2 \leq B$ *given* $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$*.*

Definition 2.2. *The LWE problem* $\mathsf{LWE}_{n,q,\chi}$*, where* n *is a dimension,* q *is a modulus and* χ *is a distribution over* $\mathbb{Z}$*, asks a computationally-bounded adversary* $\mathcal{A}$ *to distinguish between* $m = \mathsf{poly}(n)$ *samples drawn from one of the two distributions:*

1. $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$ *for a secret* $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$*, a matrix* $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ *and an error* $\mathbf{e} \leftarrow \chi^m$*;*
2. $(\mathbf{A}, \mathbf{b})$ *for* $\mathbf{A} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{m \times n}\right)$ *and* $\mathbf{b} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^m\right)$*.*

Gaussian and Statistical Lemmas The discrete Gaussian distribution on a lattice $\Lambda \subset \mathbb{Z}^l$ centered around $\mathbf{c} \in \mathbb{Z}^l$ with width $\sigma > 0$ is given by the mass function

$$\mathcal{D}_{\Lambda,\mathbf{c},\sigma}(\mathbf{z}) = \frac{\exp\left(-\pi \|\mathbf{z} - \mathbf{c}\|^2 / 2\sigma^2\right)}{\sum_{\mathbf{v} \in \Lambda} \exp\left(-\pi \|\mathbf{v} - \mathbf{c}\|^2 / 2\sigma^2\right)}.$$

We write $\mathcal{D}_{\Lambda,\sigma}$ to denote the discrete Gaussian distribution centered at $\mathbf{0}$ with width σ. In the case $\Lambda = \mathbb{Z}^l$, we simply write $\mathcal{D}_\sigma^l$.

Lemma 2.1 ([3]). *For any full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$ and any $\sigma > 0$, we have* $\Pr[\|\mathbf{v}\| > \sigma\sqrt{n}] \leq 2^{-2n}$ *where* $\mathbf{v} \leftarrow \mathcal{D}_{\mathcal{L},\sigma}$.

Throughout this paper, we use of a variant of Leftover Hash Lemma.

Lemma 2.2 (Adapted from [33, Lemma 1]). *Let $q \geq 2$ be a prime and $\mathcal{D}$ be a distribution over $\mathbb{Z}_q^m$ with min-entropy k. For any $\varepsilon > 0$ and $k \geq n \log q + 2\log(1/\varepsilon) + \mathcal{O}(1)$, the statistical distance between the two distributions*

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

is at most ε.

Trapdoor Sampling We recall the notion of *gadget matrix* [59], that is a matrix of the form

$$\mathbf{G}_{n,m} = \left(\mathbf{I}_n \otimes \left(1 \;\; 2 \;\; \ldots \;\; 2^{\lceil \log q \rceil - 1}\right) \mid \mathbf{0}^{m - n\lceil \log q \rceil}\right) \in \mathbb{Z}_q^{n \times m}.$$

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, an invertible tag matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ and a dimension w such that $m > w \geq n\lceil \log q \rceil$, a $\mathbf{G}$-trapdoor of $\mathbf{A}$ with tag $\mathbf{H}$ is a small-norm matrix $\mathbf{R} \in \mathbb{Z}^{(m-w) \times w}$ such that $\mathbf{A} \cdot (\mathbf{R}^\top \mid \mathbf{I}_w)^\top = \mathbf{H} \cdot \mathbf{G}_{n,w} \bmod q$. For sufficiently large m, we can efficiently generate a near-uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a short $\mathbf{G}$-trapdoor.

Lemma 2.3 ([59]). *There exists a PPT algorithm* $\mathsf{TrapGen}$ *that takes as input $(1^n, 1^m)$, a prime modulus $q \geq 2$ such that $m = 2n\lceil \log q \rceil + \Omega(\lambda)$, and a tag $\mathbf{H} \in \mathbb{Z}^{n \times n}$ invertible modulo q. The algorithm outputs $\mathbf{A} = \left(\overline{\mathbf{A}} \mid \mathbf{H}\mathbf{G}_{n,n\lceil \log q \rceil} - \overline{\mathbf{A}}\mathbf{R}\right) \in \mathbb{Z}_q^{n \times m}$, where $\overline{\mathbf{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times (m - n\lceil \log q \rceil)})$, $\mathbf{R} \leftarrow \mathcal{U}(\{0,1\}^{(m-n\lceil \log q \rceil) \times n\lceil \log q \rceil})$.*

The matrix $\mathbf{R} \in \mathbb{Z}^{(m-n\lceil \log q \rceil) \times n\lceil \log q \rceil}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}$ with tag $\mathbf{H}$ and $\|\mathbf{R}\|_2 \leq \sqrt{n\lceil \log q \rceil (m - n\lceil \log q \rceil)} = \mathcal{O}(n \log q)$, and the distribution of $\mathbf{A}$ is within a distance at most $2^{-\Omega(\lambda)}$ to uniform.

The result of [59] states that one can sample from any discrete Gaussian and invert LWE samples provided a sufficiently short $\mathbf{G}$-trapdoor.

Lemma 2.4 (Adapted from [59, Theorem 5.1]). *Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a $\mathbf{G}$-trapdoor $\mathbf{R} \in \mathbb{Z}^{(m-w) \times w}$ of $\mathbf{A}$ with tag $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, where $w \geq n\lceil \log q \rceil$, there exist PPT algorithms* Invert *and* $\mathsf{SampleD}$ *that with overwhelming probability over all random choices, do the following:*

- *If $\mathbf{b} \in \mathbb{Z}_q^m$, the deterministic algorithm $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b})$ outputs $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}^m$. Furthermore, if $\mathbf{b} = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{s} \in \mathbb{Z}_q^n$ and $\|\mathbf{e}\| < q/\left(2\sqrt{5} \cdot \sqrt{1 + \|\mathbf{R}\|_2^2}\right)$, then $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b})$ outputs the pair $(\mathbf{s}, \mathbf{e})$.*
- *For any $\mathbf{u} \in \mathbb{Z}_q^n$ and any $\sigma \geq \mathcal{O}(1) \cdot \sqrt{1 + \|\mathbf{R}\|_2^2}$, the randomized algorithm $\mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \mathbf{u}, \sigma)$ outputs $\mathbf{v} \in \mathbb{Z}^m$ from a distribution within a statistical distance $2^{-\Omega(n)}$ to $\mathcal{D}_\sigma^m$, conditioned on $\mathbf{A} \cdot \mathbf{v} = \mathbf{u} \bmod q$.*

The security argument of our construction implicitly uses the following result.

Lemma 2.5 (Adapted from [21, Theorem 3.4]). *Let n, q, m, k, s be positive integers such that $m \geq 2n \log q$ and $s \leq k$. Let $\sigma > \omega(\sqrt{\log m})$ be a Gaussian width. For any $\mathbf{u} \in \mathbb{Z}_q^n$ and uniformly distributed $\mathbf{A} = (\mathbf{A}_1 \mid \mathbf{A}_2) \in \mathbb{Z}_q^{n \times km}$, where $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times sm}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{n \times (k-s)m}$; consider a (possibly inefficient) algorithm that: (i) samples $\mathbf{x}_2 \leftarrow \mathcal{D}_\sigma^{(k-s)m}$; (ii) samples $\mathbf{x}_1 \leftarrow \mathcal{D}_\sigma^{sm}$ conditioned on $\mathbf{A}_1\mathbf{x}_1 = \mathbf{u} - \mathbf{A}_2\mathbf{x}_2 \bmod q$; (iii) outputs $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$. Then the output $\mathbf{x}$ is distributed within a statistical distance $\mathsf{negl}(n)$ to $\mathcal{D}_\sigma^{km}$, conditioned on $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.*

Rejection Sampling The trapdoor Σ-protocol in Section 3.2 will make use of rejection sampling technique [54].

Lemma 2.6 (Adapted from [54, Theorem 4.6]). *Let $V \subseteq \mathbb{Z}^l$ of which all elements have norms at most T, let σ be a real number such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to [0, 1]$ be a probability distribution. Then, there exists a real number M such that the distribution of the following algorithm $\mathcal{A}$:*

1. *$\mathbf{v} \leftarrow h; \mathbf{z} \leftarrow \mathcal{D}_{\sigma,\mathbf{v}}^l$*
2. *output $(\mathbf{z}, \mathbf{v})$ with probability $\min\left(\frac{\mathcal{D}_\sigma^l(\mathbf{z})}{M \cdot \mathcal{D}_{\sigma,\mathbf{v}}^l(\mathbf{z})}, 1\right)$*

is within statistical distance $\frac{2^{-\omega(\log m)}}{M}$ from the output distribution produced by the following algorithm $\mathcal{F}$:

1. *$\mathbf{v} \leftarrow h; \mathbf{z} \leftarrow \mathcal{D}_\sigma^l$*
2. *output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.*

Moreover, the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$. More concretely, if $\sigma = \alpha T$ for any positive α, then $M = e^{12/\alpha + 1/(2\alpha^2)}$, the output of $\mathcal{A}$ is within statistical distance $2^{-100}/M$ of the output of $\mathcal{F}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-100}}{M}$.

2.3 Trapdoor Σ-Protocol

We recall the notion of Σ-protocols in the *common reference string* (CRS) model.

Definition 2.3 (Adapted from [19]). *Consider a language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ associated with two NP relations $\mathcal{R}_{\text{zk}}, \mathcal{R}_{\text{snd}}$. A 3-move interactive proof system $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ in the common reference string model is a Σ-protocol for $\mathcal{L}$ if it satisfies the following conditions:*

- *The PPT algorithm $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ on input a security parameter $\lambda \in \mathbb{N}$ outputs public parameters par.*
- *The PPT algorithm $\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathsf{info}_{\mathcal{L}})$, on input public parameters par and a language-specific information $\mathsf{info}_{\mathcal{L}}$, outputs the language-reference string $\mathsf{crs}_{\mathcal{L}}$ of the full common reference string $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$.*

- ***3-Move Form***: P *and* V *both take as inputs* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*, with* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$ *and* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(1^{\lambda}, \mathcal{L})$*, and a statement* x *and proceed as follows: (i)* P *takes as input* $w \in \mathcal{R}_{\mathrm{zk}}(x)$*, computes* $(\mathbf{a}, st) \leftarrow P(\mathsf{crs}, x, w)$ *and sends* $\mathbf{a}$ *to* V*; (ii)* V *sends back a random challenge* Chall *from the challenge space* $\mathcal{C}$*; (iii)* P *finally sends a response* $\mathbf{z} = \mathsf{P}(\mathsf{crs}, x, w, \mathbf{a}, \mathsf{Chall}, st)$ *to* V*; (iv) On input of a transcript* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$*,* V *outputs 1 or 0.*
- ***Completeness***: *If* $(x, w) \in \mathcal{R}_{\mathrm{zk}}$ *and* P *honestly computes* $(\mathbf{a}, \mathbf{z})$ *for a challenge* Chall*,* $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathsf{Chall}, \mathbf{z}))$ *outputs 1 with probability* $1 - \mathrm{negl}(\lambda)$*.*
- ***Special zero-knowledge***: *There is a PPT simulator* ZKSim *that, on input a* crs*, a statement* $x \in \mathcal{L}_{\mathrm{zk}}$ *and a challenge* $\mathsf{Chall} \in \mathcal{C}$*, outputs* $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ *such that* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *is computationally indistinguishable from a real transcript with challenge* Chall *(for* $w \in \mathcal{R}_{\mathrm{zk}}(x)$*).*
- ***Special soundness***: *for any common reference string* $\mathsf{crs} \leftarrow \mathsf{Gen}(1^{\lambda})$*, any* $x \notin \mathcal{L}_{\mathrm{snd}}$*, and any first message* $\mathbf{a}$ *sent by the prover, there is at most one challenge* $\mathsf{Chall} = f(\mathsf{crs}, x, \mathbf{a})$ *for which a valid transcript* $(\mathsf{crs}, x, \mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *exists for some third message* $\mathbf{z}$*. The function* f *is called the "bad challenge function" associated with* Π*. That is, if* $x \notin \mathcal{L}_{\mathrm{snd}}$ *and the challenge differs from the bad challenge, the verifier never accepts.*

The notion of *trapdoor* Σ*-protocol* was proposed by Canetti *et al.* [19], formulated as follows:

Definition 2.4 (Adapted from [19]). *A* Σ*-protocol* $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ *with bad challenge function* f *for a trapdoor language* $\mathcal{L} = (\mathcal{L}_{\mathrm{zk}}, \mathcal{L}_{\mathrm{snd}})$ *is a* ***trapdoor*** Σ***-protocol*** *if it satisfies the properties of Definition 2.3 and there exists PPT algorithms* $(\mathsf{TrapGen}, \mathsf{BadChallenge})$ *with the following properties:*

- $\mathsf{TrapGen}(\mathsf{par}, \mathsf{info}_{\mathcal{L}})$ *is a randomized algorithm, on input public parameters* par *and an information* $\mathsf{info}_{\mathcal{L}}$*, outputs the language dependent part* $\mathsf{crs}_{\mathcal{L}}$ *of the common reference string* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ *and a trapdoor* $\tau_{\Sigma} \in \{0,1\}^{\ell_{\tau}}$ *for some* $\ell_{\tau}(\lambda)$*.*
- $\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a})$ *on input a trapdoor* τ_{Σ}*, a CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*, an instance* x *and a first-move message* $\mathbf{a}$*, outputs a challenge* Chall*.*

We require the following:

- ***CRS Indistinguishability***: *The distributions of* $\mathsf{crs}_{\mathcal{L}}$*, output from* $\mathsf{Gen}_{\mathsf{par}}$ *and* $\mathsf{TrapGen}$ *are indistinguishable.*
- ***Correctness***: *For any* $x \notin \mathcal{L}_{\mathrm{snd}}$*; all pairs* $(\mathsf{crs}_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \mathsf{info}_{\mathcal{L}})$ *and any prover's first-move message* $\mathbf{a}$*, we have that*

$$\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a}) = f(\mathsf{crs}, x, \mathbf{a}).$$

In a trapdoor Σ-protocol, for any false statement $x \notin \mathcal{L}_{\mathrm{snd}}$ there exists *at most one* challenge that prover can response; such challenge can be efficiently computed by $\mathsf{BadChallenge}$ algorithm, given as inputs a trapdoor information τ (that may depend on the crs), a false statement $x \notin \mathcal{L}_{\mathrm{snd}}$ and a first-move message a.

2.4 Correlation Intractable Hash Functions

We provide the definition of correlation intractable hash functions following Canetti *et al.* [19]. They considered a class of a binary relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ that is unique-output, i.e. for any $x \in \mathcal{X}$, there is at most one $y \in \mathcal{Y}$ such that $(x, y) \in \mathcal{R}$. Such a relation $\mathcal{R}$ is *efficiently searchable in time* T if there is a function f efficiently computable in time T such that if $(x, y) \in \mathcal{R}$ then $f(x) = y$.

For a unique-output, efficiently searchable relation $\mathcal{R}$, a hash family $\mathcal{H}$ that is *correlation intractable* w.r.t $\mathcal{R}$ satisfies that, for a hash function h with key k generated from $\mathcal{H}$, it is computationally hard for a PPT adversary to find $(x, y) \in \mathcal{R}$, such that $h(k, x) = y$. More formally, let $\lambda \in \mathbb{N}$ be a security parameter. A hash family with input length $n(\lambda)$ and output length $m(\lambda)$ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ of keyed functions realized by efficient algorithms $(\mathsf{Gen}, \mathsf{Hash})$ where $\mathsf{Gen}(1^\lambda)$ outputs a key $k \in \{0,1\}^{s(\lambda)}$ and $\mathsf{Hash}(k, x)$ computes a hash value $h_\lambda(k, x)$.

Definition 2.5 ([19]). *For a relation ensemble $\{\mathcal{R}_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$, a hash function family $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ is $\mathcal{R}$-correlation intractable if, for any PPT adversary $\mathcal{A}$, we have*

$$\Pr[k \leftarrow \mathsf{Gen}(1^\lambda), x \leftarrow \mathcal{A}(k) : (x, h_\lambda(k, x)) \in \mathcal{R}] = \mathsf{negl}(\lambda).$$

[19] also considered a stronger definition of *somewhere statistically correlation intractable* for a hash family $\mathcal{H}$. Namely, the family $\mathcal{H}$ has two indistinguishable key generation algorithms KeyGen and $\mathsf{StatGen}$; and with high probability over the hash key k output by $\mathsf{StatGen}$, there is no $x \in \mathcal{X}$ such that $(x, h(k, x)) \in \mathcal{R}$.

Definition 2.6 ([19]). *Given a collection of relation ensemble $\mathfrak{R}$, a hash family $\mathcal{H}$ is somewhere statistically correlation intractable w.r.t. $\mathfrak{R}$ if there is an efficient fake key generation algorithm $\mathsf{StatGen}$ with the following properties:*

- $\mathsf{StatGen}(1^\lambda, \mathsf{aux})$ *takes as inputs a security parameter λ and an auxiliary input aux. It outputs a hashing key k;*
- *For any $\mathcal{R} \in \mathfrak{R}$, there is an auxiliary input $\mathsf{aux}_\mathcal{R}$ with these properties:*
  - ***Key indistinguishability**: The distributions $\{k : k \leftarrow \mathsf{Gen}(1^\lambda)\}$ and $\{k : k \leftarrow \mathsf{StatGen}(1^\lambda, aux_\mathcal{R})\}$ are computationally indistinguishable.*
  - ***Statistical Correlation Intractability**: With high probability over the choice of $k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_\mathcal{R})$, no pair $(k, h(k, x))$ satisfies $\mathcal{R}$, i.e. we have*

$$\Pr_{k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_\mathcal{R})}[\exists x \in \{0,1\}^{n(\lambda)} : (x, h(k, x)) \in \mathcal{R}] \leq 2^{-\Omega(\lambda)}.$$

- *Universality: for any $\lambda \in \mathbb{N}$, input $x \in \{0,1\}^n$, and output $y \in \{0,1\}^m$, we have $\Pr[h(k, x) = y \mid k \leftarrow \mathsf{Gen}(1^\lambda)] = 2^{-m}$.*
- *Programmability: there exists an efficient algorithm $\mathsf{Sample}(1^\lambda, x, y)$ that samples from the conditional distribution $\{k \leftarrow \mathsf{Gen}(1^\lambda) \mid h(k, x) = y\}$.*

Peikert and Shiehian [64] provided a construction of somewhere statistically correlation intractable hash functions for efficiently searchable relations. Their construction is based on the GSW fully homomorphic encryption scheme [32] and relies on LWE with a polynomial approximation factor.

2.5 Group Signature

We use the syntax and security definitions of group signatures of [5]. A group signature scheme $\mathsf{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ consists of a tuple of four PPT algorithms:

- $\mathsf{KeyGen}(1^\lambda, 1^N)$ takes 1^λ and 1^N as inputs, where $\lambda \in \mathbb{N}$ is the security parameter, and $N \in \mathbb{N}$ is the maximum number of group members. It returns a tuple $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ where gpk is the *group public key*, gmsk is the group manager secret key, and gsk is an N-vector of secret keys, in which the component $\mathsf{gpk}[j]$ is the signing key of the j-th user, for $j \in [N]$.
- $\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$ takes the group public key gpk, a signing key $\mathsf{gsk}[j]$ and a message $M \in \{0,1\}^*$ as inputs. It outputd is a signature $\Sigma \in \{0,1\}^*$ on M under gpk.
- $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$ is deterministic and takes the group public key gpk, a message M and a signature Σ. It outputs either 0 (reject) or 1 (accept).
- $\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \Sigma)$ is deterministic and takes as inputs the group public key gpk, the group manager secret key gmsk, a message M and a valid group signature Σ w.r.t. gpk. It returns an index $j \in [N]$ or a special symbol $\perp$ indicating failure.

Correctness states that for all (λ, N), all $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$, all indexes $j \in [N]$ and messages $M \in \{0,1\}^*$:

$$\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = 1;$$
$$\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = j,$$

with probability negligibly close to 1 over the random coins of KeyGen and Sign.

The security requirement of group signatures is discussed in Appendix A.1.

3 A Trapdoor Σ-Protocol for Linear Relation and Dual-Regev Ciphertexts

3.1 Overview

As a building block of the group signature scheme in Section 4, we construct an NIZK proving statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l}$ such that

$$\mathbf{t} = \mathbf{R} \cdot \mathbf{x} \bmod q, \tag{3}$$

$$\forall i \in [2] : \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{x} \end{pmatrix} \bmod q, \tag{4}$$

where $\mathbf{x} \in \mathbb{Z}^l$; $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}_q^n$; $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^{m+l}$ satisfying $\|\mathbf{x}\|_2 \leq B$; $\|\mathbf{e}_1\|_2, \|\mathbf{e}_2\|_2 \leq B_{\mathbf{e}}$. Equivalently, the NIZK proves the existence of a short witness $\mathbf{x}$ satisfying a linear relation $\mathbf{R} \cdot \mathbf{x} = \mathbf{t} \bmod q$, and that two dual-Regev ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ encrypt $\mathbf{x}$.

We start with constructing a trapdoor Σ-protocol proving (3) and (4), then convert it into an NIZK via the compiler of [45]. As (3) and (4) are linear relations with some norm constraints over the witnesses, it suffices to apply Lyubashevsky's Σ-protocol [53,54]. In this Σ-protocol, the prover produces a response $(\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{e}_1}, \mathbf{z}_{\mathbf{r}_2}, \mathbf{z}_{\mathbf{e}_2})$ satisfying

$$\begin{aligned} \mathbf{a} + c \cdot \mathbf{t} &= \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}} \bmod q, \\ \forall i \in [2] : \mathbf{v}_i + c \cdot \mathbf{c}_i &= \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rceil \cdot \mathbf{z}_{\mathbf{x}} \end{pmatrix} \bmod q; \end{aligned}$$

where c is a verifier's challenge and $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2)$ is the prover's first message. Thus, assuming there is a trapdoor for randomness recovery in dual-Regev encryption scheme and that the parameters are set appropriately, then given a "ciphertext" $\mathbf{v}_i + c \cdot \mathbf{c} \bmod q$, we can recover the pair of encryption randomness and plaintext $(\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i})$ to decide if it is a valid response or not. This serves as the key idea for designing the bad challenge finding algorithm.

We remark that in the proposed Σ-protocol, the matrix $\mathbf{R}$ is allowed to be a part of prover's statement, where the dual-Regev public keys $\{(\mathbf{U}_i, \mathbf{V}_i)\}_{i \in [2]}$ are part of the language reference string and are generated along with the CRS.

3.2 Description of the Protocol

We give the details of the trapdoor Σ-protocol proving (3) and (4) as follows:

- $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ on input a security parameter $\lambda \in \mathbb{N}$, choose large prime modulus $q \in \mathsf{poly}(\lambda)$, dimensions $n = \mathcal{O}(\lambda), m = 2n\lceil \log q \rceil + \Omega(\lambda)$ and $l = \mathsf{poly}(\lambda)$. Output $\mathsf{par} = \{q, n, m, l\}$.
- $\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathsf{info}_{\mathcal{L}})$ takes as input public parameters par, a language-dependent information $\mathsf{info}_{\mathcal{L}}$ specifying a norm bound $B > 0$. The algorithm performs the below steps:
  1. For $i \in \{1, 2\}$; sample $\mathbf{U}_i \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$, $\mathbf{S}_i \leftarrow \{0, 1\}^{m \times l}$ and let $\mathbf{V}_i = \mathbf{U}_i \cdot \mathbf{S}_i \bmod q \in \mathbb{Z}_q^{n \times l}$. The pair $(\mathbf{U}_i, \mathbf{V}_i)$ defines the public-key of dual-Regev encryption scheme and the matrix $\mathbf{S}_i$ defines the corresponding decryption key;
  2. Choose a distribution χ over $\mathbb{Z}$ for sampling encryption randomness. We require that there is a bound $B_{\mathbf{e}} = \mathsf{poly}(\lambda)$ such that for $\mathbf{e} \leftarrow \chi^{m+l}$, we have that $\|\mathbf{e}\|_2 \leq B_{\mathbf{e}}$ except with probability $\mathsf{negl}(\lambda)$;
  3. Choose a positive integer $C = \mathsf{poly}(\lambda)$ defining the challenge space $\mathcal{C} = \{0, 1, \ldots, C\}$ and chooses a Gaussian width $\alpha > 0$ and a small parameter $\gamma \geq 1$ such that

$$\alpha = C \cdot \sqrt{B^2 + 2B_{\mathbf{e}}^2} \cdot \gamma \tag{5}$$

and set $M = \exp(12/\gamma + 1/(2\gamma^2))$. The parameter M dictates the rejection rate of the protocol.

4. Choose an integer $K = \mathsf{poly}(\lambda)$ that satisfies

$$\alpha\sqrt{l} < K/2, \quad \alpha\sqrt{m+l} < q/4K. \tag{6}$$

and

$$\alpha\sqrt{m+l} \cdot \sqrt{1 + n\lceil \log q \rceil (m - n\lceil \log q \rceil)} < q/2\sqrt{5}. \tag{7}$$

Let

$$\mathsf{crs}_{\mathcal{L}} = \left(\{(\mathbf{U}_i, \mathbf{V}_i)\}_{i\in\{1,2\}}, B, \chi, B_{\mathbf{e}}, C, \alpha, K\right),$$

and output $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$. The decryption keys $\mathbf{S}_1, \mathbf{S}_2$ are also kept for later use in the group signature, and play no role in the protocol. The string $\mathsf{crs}_{\mathcal{L}}$ defines the language $\mathcal{L} = (\mathcal{L}_{\mathrm{zk}}, \mathcal{L}_{\mathrm{snd}})$, where

$$\begin{aligned}
\mathcal{L}_{\mathrm{zk}} = \{&(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l} : \\
&\exists(\mathbf{x}, \mathbf{r}_1, \mathbf{e}_1, \mathbf{r}_2, \mathbf{e}_2) \in \mathbb{Z}^l \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \text{ s.t.} \\
&(\mathbf{t} = \mathbf{R}\cdot\mathbf{x} \bmod q) \wedge \left(\forall i \in [2] : \mathbf{c}_i = \begin{pmatrix}\mathbf{U}_i^\top \\ \mathbf{V}_i^\top\end{pmatrix} \cdot \mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix}\mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{x}\end{pmatrix} \bmod q\right) \\
&\wedge \|\mathbf{x}\|_2 \le B \wedge \|\mathbf{e}_1\|_2 \le B_{\mathbf{e}} \wedge \|\mathbf{e}_2\|_2 \le B_{\mathbf{e}}\},
\end{aligned} \tag{8}$$

and

$$\begin{aligned}
\mathcal{L}_{\mathrm{snd}} = \{&(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l} : \\
&\exists(\bar{c}, \overline{\mathbf{x}}, \overline{\mathbf{r}}_1, \overline{\mathbf{e}}_1, \overline{\mathbf{r}}_2, \overline{\mathbf{e}}_2) \in [C] \times \mathbb{Z}^l \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \text{ s.t.} \\
&(\bar{c}\cdot\mathbf{t} = \mathbf{R}\cdot\overline{\mathbf{x}} \bmod q) \wedge \left(\forall i \in [2] : \bar{c}\cdot\mathbf{c}_i = \begin{pmatrix}\mathbf{U}_i^\top \\ \mathbf{V}_i^\top\end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix}\mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \overline{\mathbf{x}}\end{pmatrix} \bmod q\right) \\
&\wedge \|\overline{\mathbf{x}}\|_2 \le 2\alpha\sqrt{l} \wedge \|\overline{\mathbf{e}}_1\|_2 \le 2\alpha\sqrt{m+l} \wedge \|\overline{\mathbf{e}}_2\|_2 \le 2\alpha\sqrt{m+l}\},
\end{aligned} \tag{9}$$

– $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L})$ is identical to $\mathsf{Gen}_{\mathcal{L}}$, the difference is that it samples the matrix $\mathbf{U}_i \in \mathbb{Z}_q^{n\times m}$ along with a $\mathbf{G}$-trapdoor $\mathbf{T}_i \in \{0,1\}^{(m-n\lceil\log q\rceil)\times n\lceil\log q\rceil}$, by using the algorithm $\mathsf{TrapGen}$ of Lemma 2.3. It sets the trapdoor $\tau = (\mathbf{T}_1, \mathbf{T}_2)$. We remark that for $i \in \{1,2\}$, we have $\|\mathbf{T}_i\|_2 \le \sqrt{n\lceil\log q\rceil(m - n\lceil\log q\rceil)} = \mathcal{O}(n\log q)$.

– $\mathsf{P}(\mathsf{crs}, (\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2), (\mathbf{x}, \mathbf{r}_1, \mathbf{e}_1, \mathbf{r}_2, \mathbf{e}_2))$ and $\mathsf{V}(\mathsf{crs}, (\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2))$ take as common input a CRS $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$, a statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l}$. The prover P takes as private input a witness $(\mathbf{x}, \mathbf{r}_1, \mathbf{e}_1, \mathbf{r}_2, \mathbf{e}_2) \in \mathbb{Z}^l \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l}$ and interacts with verifier V as depicted in Figure 1.

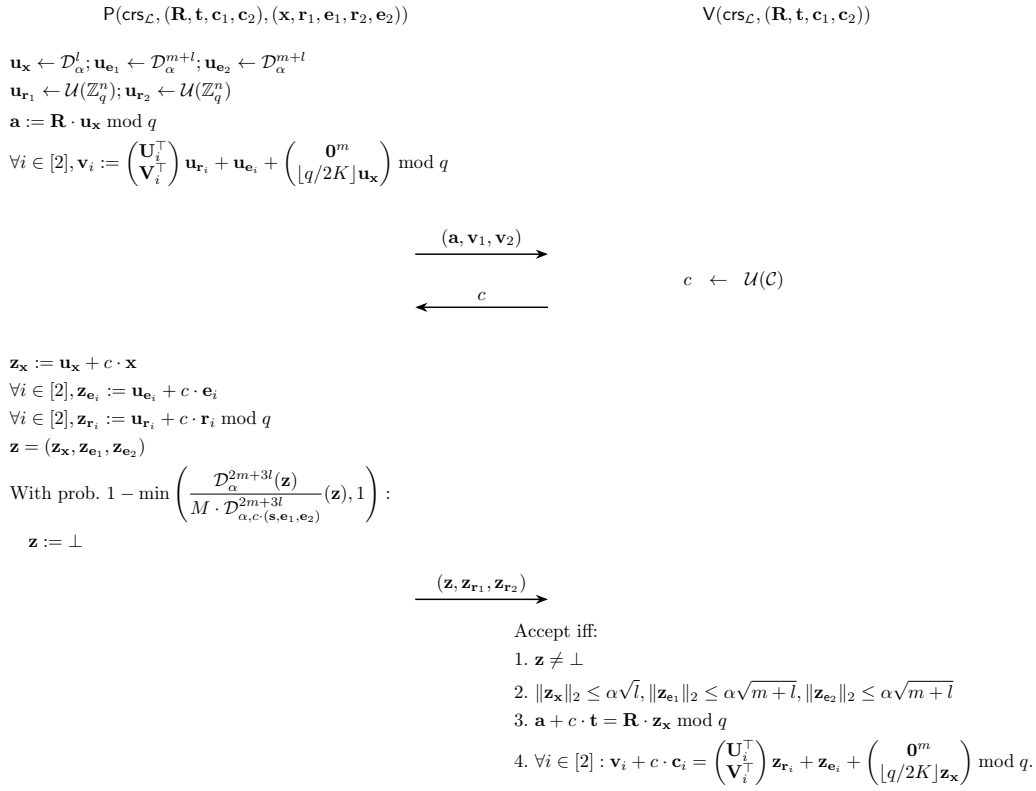

Fig. 1. Σ-protocol proving $\mathcal{L}$

– $\mathsf{BadChallenge}(\tau, \mathsf{crs}, (\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2), (\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2))$ takes as input a trapdoor $\tau = (\mathbf{T}_1, \mathbf{T}_2) \in \left(\{0,1\}^{(m-n\lceil \log q \rceil) \times n\lceil \log q \rceil}\right)^2$, a $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$, a statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l}$ and a first-move message $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+l} \times \mathbb{Z}_q^{m+l}$. Then for $i \in \{1, 2\}$, the algorithm does the following

1. For each $c_i \in \mathcal{C} = \{0, 1, \ldots, C\}$, compute $\mathbf{v}_i + c \cdot \mathbf{c}_i \bmod q = \left(\mathbf{u}_i^{(1)}, \mathbf{u}_i^{(2)}\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^l$. Then decide if there exist $\mathbf{z}_{\mathbf{r}_i} \in \mathbb{Z}_q^n$ and $\mathbf{z}_{\mathbf{e}_i}^{(1)} \in \mathbb{Z}^m$ such that

$$\mathbf{u}_i^{(1)} = \mathbf{U}_i^\top \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i}^{(1)} \bmod q, \tag{10}$$

and $\left\|\mathbf{z}_{\mathbf{e}_i}^{(1)}\right\|_2 \leq \alpha\sqrt{m+l}$. This is done using algorithm Invert of Lemma 2.4. Note that, by the conditions in (7), algorithm Invert correctly recovers $\mathbf{z}_{\mathbf{r}_i}$ and $\mathbf{z}_{\mathbf{e}_i}^{(1)}$ if they exist.

2. For each tuple $(c_i, \mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i}^{(1)})$ found in the above step, test if there exist $\mathbf{z}_{\mathbf{e}_i}^{(2)} \in \mathbb{Z}^l$ and $\mathbf{z}_{\mathbf{x}}^{(i)} \in \mathbb{Z}^l$ such that

$$\mathbf{u}_i^{(2)} - \mathbf{V}_i^\top \cdot \mathbf{z}_{\mathbf{r}_i} \bmod q = \lfloor q/2K \rfloor \cdot \mathbf{z}_{\mathbf{x}}^{(i)} + \mathbf{z}_{\mathbf{e}_i}^{(2)}, \tag{11}$$

and that $\left\|\mathbf{z}_{\mathbf{x}}^{(i)}\right\|_2 \leq \alpha\sqrt{l}$, $\left\|\mathbf{z}_{\mathbf{e}_i}^{(2)}\right\|_2 \leq \alpha\sqrt{m+l}$. From (6), we have $\alpha\sqrt{l} < K/2$ and $\alpha\sqrt{m+l} < q/4K$. Therefore the pair $(\mathbf{z}_{\mathbf{x}}^{(i)}, \mathbf{z}_{\mathbf{e}_i}^{(2)})$ is unique if it exists, and can be recovered by using Euclidean algorithm.

3. For each tuple $(c_i, \mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i}^{(1)}, \mathbf{z}_{\mathbf{e}_i}^{(2)}, \mathbf{z}_{\mathbf{x}}^{(i)})$ found, parse $\mathbf{z}_{\mathbf{e}_i} = \left(\mathbf{z}_{\mathbf{e}_i}^{(1)}, \mathbf{z}_{\mathbf{e}_i}^{(2)}\right) \in \mathbb{Z}^{m+l}$. Decide if $\|\mathbf{z}_{\mathbf{e}_i}\|_2 \leq \alpha \cdot \sqrt{m+l}$ and if

$$\mathbf{a} + c \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}}^{(i)} \bmod q. \tag{12}$$

Let $\mathcal{C}_i$ be the set of all candidates $(c_i, \mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i}^{(1)}, \mathbf{z}_{\mathbf{e}_i}^{(2)}, \mathbf{z}_{\mathbf{x}}^{(i)})$ found after the three steps above.

Finally, the algorithm decides if there exist $(c_1, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{e}_1}^{(1)}, \mathbf{z}_{\mathbf{e}_1}^{(2)}, \mathbf{z}_{\mathbf{x}}^{(1)}) \in \mathcal{C}_1$ and $(c_2, \mathbf{z}_{\mathbf{r}_2}, \mathbf{z}_{\mathbf{e}_2}^{(1)}, \mathbf{z}_{\mathbf{e}_2}^{(2)}, \mathbf{z}_{\mathbf{x}}^{(2)}) \in \mathcal{C}_2$ such that $c_1 = c_2$ and $\mathbf{z}_{\mathbf{x}}^{(1)} = \mathbf{z}_{\mathbf{x}}^{(2)}$. Return $\perp$ if they do not exist, otherwise return $c = c_1 = c_2$.

Proposition 3.1. *The protocol in* ***Fig.*** *1 is a trapdoor Σ-protocol for the language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ specified by (8) and (9).*

Proof. We show that the conditions in Definition 2.4 are satisfied.

CRS indistinguishability. The distribution of $(\mathbf{U}_1, \mathbf{U}_2)$ output by $\mathsf{TrapGen}$ is at a distance at most $2^{-\Omega(\lambda)}$ to uniform (Lemma 2.3). Moreover, as $m = 2n\lceil \log q \rceil + \Omega(\lambda)$, Lemma 2.2 applied column-wise shows that $(\mathbf{U}_i, \mathbf{V}_i)$ is statistically close to uniform in both cases, where $\mathbf{V}_i = \mathbf{U}_i \cdot \mathbf{S}_i \bmod q$. It follows that the CRS distributions of $\mathsf{TrapGen}$ and $\mathsf{Gen}_{\mathcal{L}}$ are statistically indistinguishable.

Completeness. From Lemma 2.6, the probability that $\mathbf{z} \neq \perp$ is at least $(1 - 2^{-100})/M$. Moreover, as $(\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{e}_1}, \mathbf{z}_{\mathbf{e}_2})$ follows a distribution statistically close to $\mathcal{D}_\alpha^{3m+2l}$; we have that $\|\mathbf{z}_{\mathbf{x}}\|_2 \leq \alpha\sqrt{l}$ and $\|\mathbf{z}_{\mathbf{e}_1}\|_2, \|\mathbf{z}_{\mathbf{e}_2}\|_2 \leq \alpha\sqrt{m+l}$ with overwhelming probability (Lemma 2.1). Finally, as $\mathbf{z}_{\mathbf{x}} = \mathbf{u}_{\mathbf{x}} + c \cdot \mathbf{x}$, $\mathbf{z}_{\mathbf{e}_i} = \mathbf{u}_{\mathbf{e}_i} + c \cdot \mathbf{e}_i$ and $\mathbf{z}_{\mathbf{r}_i} = \mathbf{u}_{\mathbf{r}_i} + c \cdot \mathbf{r}_i \bmod q$, we can verify that

$$\mathbf{a} + c \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}} \bmod q,$$
$$\forall i \in [2] : \mathbf{v}_i + c \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}_{\mathbf{x}} \end{pmatrix} \bmod q.$$

Special soundness. It suffices to prove that for a statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^n \times (\mathbb{Z}_q^{m+l})^2$ and a first-move message $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_q^n \times (\mathbb{Z}_q^{m+l})^2$, if there are two distinct challenges $c, c' \in \mathcal{C} = \{0, 1, \ldots, C\}$ with corresponding valid responses

$$(\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{e}_1}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{e}_2}, \mathbf{z}_{\mathbf{r}_2}) \in \mathbb{Z}^l \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n$$
$$(\mathbf{z}'_{\mathbf{x}}, \mathbf{z}'_{\mathbf{e}_1}, \mathbf{z}'_{\mathbf{r}_1}, \mathbf{z}'_{\mathbf{e}_2}, \mathbf{z}'_{\mathbf{r}_2}) \in \mathbb{Z}^l \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n \times \mathbb{Z}^{m+l} \times \mathbb{Z}_q^n;$$

then $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathcal{L}_{\text{snd}}$ (as specified by (9)). Note that we have

$$\mathbf{a} + c \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}} \bmod q \tag{13}$$

$$\forall i \in [2] : \mathbf{v}_i + c \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}_{\mathbf{x}} \end{pmatrix} \bmod q; \tag{14}$$

and

$$\mathbf{a} + c' \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}'_{\mathbf{x}} \bmod q, \tag{15}$$

$$\forall i \in [2] : \mathbf{v}_i + c' \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}'_{\mathbf{r}_i} + \mathbf{z}'_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}'_{\mathbf{x}} \end{pmatrix} \bmod q. \tag{16}$$

Let $\overline{c} = c - c'$, $\overline{\mathbf{x}} = \mathbf{z}_{\mathbf{x}} - \mathbf{z}'_{\mathbf{x}}$; and for $i \in [2]$ let $\overline{\mathbf{e}}_i = \mathbf{z}_{\mathbf{e}_i} - \mathbf{z}'_{\mathbf{e}_i}$ and $\overline{\mathbf{r}}_i = \mathbf{z}_{\mathbf{r}_i} - \mathbf{z}'_{\mathbf{r}_i} \bmod q$. It follows that $\|\overline{\mathbf{x}}\| \leq 2\alpha\sqrt{l}, \|\overline{\mathbf{e}}_i\| \leq 2\alpha\sqrt{m+l}$ and $\overline{c} \in [-C, C] \backslash \{0\}$. By negating $(\overline{c}, \overline{\mathbf{x}}, \overline{\mathbf{r}}_i, \overline{\mathbf{e}}_i)$ if needed, which preserves all the bounds, we may assume that $\overline{c} \in [C]$. By subtracting (13) and (15), and subtracting (14) and (16), we obtain

$$\overline{c} \cdot \mathbf{t} = \mathbf{R} \cdot \overline{\mathbf{x}} \bmod q, \tag{17}$$

$$\forall i \in [2] : \overline{c} \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \overline{\mathbf{x}} \end{pmatrix} \bmod q. \tag{18}$$

Therefore, (17) and (18) imply that $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathcal{L}_{\text{snd}}$. In other words, if $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \notin \mathcal{L}_{\text{snd}}$, for any first-move message $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2)$ there is at most one challenge admitting a valid response.

Correctness of BadChallenge. For a statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \notin \mathcal{L}_{\text{snd}}$ and a first-move message $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_q^n \times (\mathbb{Z}_q^{m+l})^2$, there can be at most one challenge c admitting a response $(\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{e}_1}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{e}_2}, \mathbf{z}_{\mathbf{r}_2})$ satisfying

$$\mathbf{a} + c \cdot \mathbf{t} = \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}} \bmod q, \tag{19}$$

$$\forall i \in [2] : \mathbf{v}_i + c \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}_{\mathbf{x}} \end{pmatrix}; \tag{20}$$

where $\|\mathbf{z}_{\mathbf{x}}\|_2 \leq \alpha \cdot \sqrt{l}$ and $\|\mathbf{z}_{\mathbf{e}_i}\|_2 \leq \alpha \cdot \sqrt{m+l}$. Note that, if there is a tuple $(c, (\mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i}^{(1)}, \mathbf{z}_{\mathbf{e}_i}^{(2)})_{i \in [2]}, \mathbf{z}_{\mathbf{x}})$ surviving all the steps of BadChallenge, then it satisfies the equations (10), (11) and (12). These equations imply $\big(c, ((\mathbf{z}_{\mathbf{r}_i}, \mathbf{z}_{\mathbf{e}_i})_{i \in [2]}, \mathbf{z}_{\mathbf{x}})\big)$ is a pair of challenge and response satisfying (19) and (20), where $\mathbf{z}_{\mathbf{e}_i} = (\mathbf{z}_{\mathbf{e}_i}^{(1)}, \mathbf{z}_{\mathbf{e}_i}^{(2)})$. Therefore, running BadChallenge on $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \notin \mathcal{L}_{\text{snd}}$ and any first-move message $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_q^n \times (\mathbb{Z}_q^{m+l})^2$ should return the unique bad challenge if it exists. Furthermore, the steps of BadChallenge runs in polynomial time and correctness then follows.

Special zero-knowledge. We construct a simulator that takes as input crs, a statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^{n \times l} \times \mathbb{Z}_q^n \times (\mathbb{Z}_q^{m+l})^2$, a challenge $c \in \mathcal{C} = \{0, 1, \ldots, C\}$ and returns a pair $((\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2), (\mathbf{z}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{r}_2}))$ of prover's initial message and response. The simulator proceeds in the following steps:

1. Sample $\mathbf{z} \leftarrow \mathcal{D}_\alpha^{2m+3l}$, $\mathbf{z}_{\mathbf{r}_1} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{z}_{\mathbf{r}_2} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$;
2. Parse $\mathbf{z} = (\mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{e}_1}, \mathbf{z}_{\mathbf{e}_2}) \in \mathbb{Z}^l \times (\mathbb{Z}^{m+l})^2$ and compute

$$\mathbf{a} = \mathbf{R} \cdot \mathbf{z}_{\mathbf{x}} - c \cdot \mathbf{t} \bmod q \in \mathbb{Z}_q^n$$
$$\forall i \in [2] : \mathbf{v}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{z}_{\mathbf{r}_i} + \mathbf{z}_{\mathbf{e}_i} + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \mathbf{z}_{\mathbf{x}} \end{pmatrix} - c \cdot \mathbf{c}_i \bmod q \in \mathbb{Z}_q^{m+l}.$$

3. Output $((\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2), (\mathbf{z}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{r}_2}))$.

In a real execution of the protocol, the distribution of $\mathbf{z}$ is within statistical distance $2^{-100}/M$ from $\mathcal{D}_\alpha^{2m+3l}$ (Lemma 2.6). Moreover, the response $\mathbf{z}_{\mathbf{r}_i} = \mathbf{u}_{\mathbf{r}} + c \cdot \mathbf{r} \bmod q$ is distributed uniformly over $\mathbb{Z}_q^n$ since the mask vector $\mathbf{u}_{\mathbf{r}_i}$ is uniform. Since $(\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2)$ is uniquely determined from the challenge c and response $(\mathbf{z}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{r}_2})$, the distribution of $((\mathbf{a}, \mathbf{v}_1, \mathbf{v}_2), (\mathbf{z}, \mathbf{z}_{\mathbf{r}_1}, \mathbf{z}_{\mathbf{r}_2}))$ output by the simulator is statistically close to a non-aborting transcript from a real protocol execution with c. □

We remark that the Σ-protocol of **Fig.** 1 admits a non-negligible completeness error and is not fully compatible with Definition 2.4. Nevertheless, in the resulting NIZK from Fiat-Shamir transformation, a prover can simply restart any aborting instance during parallel execution.

Boosting soundness error. The trapdoor Σ-protocol of Section 3.2 has an inverse-polynomial soundness error and thus requires $\kappa = \Theta(\lambda / \log \lambda)$ parallel executions. Namely, the prover computes κ first messages $(\mathbf{a}^{(i)}, \mathbf{v}_1^{(i)}, \mathbf{v}_2^{(i)})_{i \in [\kappa]}$, and after receiving the challenges $\overline{\mathbf{c}} = (c_i)_{i \in [\kappa]}$ it computes the responses $(\mathbf{z}^{(i)}, \mathbf{z}_{\mathbf{r}_1}^{(i)}, \mathbf{z}_{\mathbf{r}_2}^{(i)})_{i \in [\kappa]}$, where $\mathbf{z}^{(i)} = (\mathbf{z}_{\mathbf{x}}^{(i)}, \mathbf{z}_{\mathbf{e}_1}^{(i)}, \mathbf{z}_{\mathbf{e}_2}^{(i)})$. To reduce the rejection rate, the prover performs the rejection sampling jointly over all κ parallel executions. Let $\overline{\mathbf{z}} = (\mathbf{z}^{(i)})_{i \in [\kappa]}$ and $\overline{\mathbf{c}} \otimes (\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2) = (c_i \cdot (\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2))_{i \in [\kappa]}$. The prover outputs the responses with probability

$$\min \left(\frac{\mathcal{D}_\alpha^{\kappa(2m+3l)}(\overline{\mathbf{z}})}{M \cdot \mathcal{D}_{\alpha, \overline{\mathbf{c}} \otimes (\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2)}^{\kappa(2m+3l)}(\overline{\mathbf{z}})}, 1 \right).$$

Since $\|\overline{\mathbf{c}} \otimes (\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2)\|_2 \leq \sqrt{\kappa} \cdot C \cdot \sqrt{B^2 + 2B_{\mathbf{e}}^2}$, Lemma 2.6 requires the Gaussian width

$$\alpha = \gamma \cdot \sqrt{\kappa} \cdot C \cdot \sqrt{B^2 + 2B_{\mathbf{e}}^2}$$

for a parameter $\gamma \geq 1$, in place of the width in (5), with $M = \exp(12/\gamma + 1/(2\gamma^2))$. When applying the compiler of [45], if the joint rejection test rejects, the prover discards the entire proving attempt and restarts with fresh randomness, until the joint rejection test accepts.

4 A Group Signature Scheme with CCA-Full Anonymity

4.1 The Underlying NIZK Argument System

We presented an NIZK argument system serving as a building block in our group signature scheme with CCA-anonymity. Looking ahead, in the scheme presented

in Section 4.2, signers need to prove in zero-knowledge statements defined by the following language

$$\begin{aligned}\mathcal{L}_{\mathsf{zk}}^{\mathsf{cca}} = \{&(\mathbf{b},\mathbf{d},\mathbf{t},\mathbf{c}_1,\mathbf{c}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{2m+k+\ell} \times \mathbb{Z}_q^{2m+k+\ell} : \\ &\exists (\mathbf{s}, \mathsf{id}, \tau_{\mathsf{id}}, \{(\mathbf{r}_i, \mathbf{e}_i)\}_{i\in\{1,2\}}) \in \mathbb{Z}^m \times [N] \times [q_{\mathsf{tag}}] \times \left(\mathbb{Z}_q^n \times \mathbb{Z}^{2m+k+\ell}\right)^2 \text{ s.t.} \\ &\quad \mathbf{t} = \mathbf{A}\cdot\mathbf{s} + \mathsf{id}\cdot\mathbf{b} + \tau_{\mathsf{id}}\cdot\mathbf{d} \bmod q \\ &\quad \wedge \forall i \in [2] : \mathbf{c}_i = \begin{pmatrix}\mathbf{U}_i^\top \\ \mathbf{V}_i^\top\end{pmatrix}\cdot\mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix}\mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}}))\end{pmatrix} \bmod q \\ &\quad \wedge \|\mathbf{s}\|_2 \le B_{\mathbf{s}} \wedge \|\mathbf{e}_1\|_2 \le B_{\mathbf{e}} \wedge \|\mathbf{e}_2\|_2 \le B_{\mathbf{e}}\},\end{aligned}$$

where:

- The matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$; $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{Z}_q^{n\times m}$; $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{Z}_q^{n\times(m+k+\ell)}$ are parts of group public-key; here $(\mathbf{U}_1, \mathbf{V}_1)$ and $(\mathbf{U}_2, \mathbf{V}_2)$ are public keys of dual-Regev encryption;
- $\mathbf{b}, \mathbf{d}, \mathbf{t} \in \mathbb{Z}_q^n$ are vectors that can be computed from a user's signature;
- The positive integer $N = 2^k - 1$ denotes the maximum size of the group;
- The integer $\mathsf{id} \in [N]$ is a user's unique identifier and $\mathsf{bin}(\mathsf{id}) \in \{0,1\}^k$ is the corresponding binary representation.
- The positive integer $q_{\mathsf{tag}} < q/2$ specifies the tag space in JRS signature [36]. We require that $q_{\mathsf{tag}} = 2^\ell - 1$;
- The integer $\tau_{\mathsf{id}} \in [q_{\mathsf{tag}}]$ is the tag in the JRS signature and $\mathsf{bin}(\tau_{\mathsf{id}}) \in \{0,1\}^\ell$ is its binary representation.

In essence, the relation $\mathcal{L}_{\mathsf{zk}}^{\mathsf{cca}}$ captures the well-formedness of a signature, which requires:

- User id has a short vector $\mathbf{s}$ and a tag τ_{id} satisfying a linear equation $\mathbf{t} = \mathbf{A}\cdot\mathbf{s} + \mathsf{id}\cdot\mathbf{b} + \tau_{\mathsf{id}}\cdot\mathbf{d} \bmod q$;
- Two dual-Regev ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ encrypt a same plaintext $(\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}}))$, with respective randomnesses $(\mathbf{r}_1, \mathbf{e}_1)$ and $(\mathbf{r}_2, \mathbf{e}_2)$

We construct a trapdoor Σ-protocol proving $\mathcal{L}_{\mathsf{zk}}^{\mathsf{cca}}$ from the blueprint provided in Section 3.2. Such a Σ-protocol is obtained by rewriting the constraints defining $\mathcal{L}_{\mathsf{zk}}^{\mathsf{cca}}$ as

$$\begin{aligned}\mathbf{t} &= (\mathbf{A} \mid \mathbf{F} \mid \mathbf{H}) \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \bmod q, \\ \forall i \in [2] : \mathbf{c}_i &= \begin{pmatrix}\mathbf{U}_i^\top \\ \mathbf{V}_i^\top\end{pmatrix}\cdot\mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix}\mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}}))\end{pmatrix} \bmod q\end{aligned} \tag{21}$$

where $\mathbf{F} = \left(\mathbf{b} \mid 2\cdot\mathbf{b} \mid \cdots \mid 2^{k-1}\cdot\mathbf{b}\right) \in \mathbb{Z}_q^{n\times k}$ and $\mathbf{H} = \left(\mathbf{d} \mid 2\cdot\mathbf{d} \mid \cdots \mid 2^{\ell-1}\cdot\mathbf{d}\right) \in \mathbb{Z}_q^{n\times\ell}$. The prover then proves the existence of $\mathbf{s} \in \mathbb{Z}^m$; $\mathsf{bin}(\mathsf{id}) \in \{0,1\}^k$; $\mathsf{bin}(\tau_{\mathsf{id}}) \in \{0,1\}^\ell$; $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}_q^n$ and $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^{2m+k+\ell}$ satisfying (21), using the trapdoor Σ-protocol in Section 3.2. Finally, we convert the Σ-protocol to a one-time simulation sound NIZK by the compiler of [45].

Proposition 4.1. *Let λ be a security parameter; $q \in \mathsf{poly}(\lambda)$ be a large prime modulus; $n = \mathcal{O}(\lambda)$; $m = 2n\lceil \log q \rceil + \Omega(\lambda)$; $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; $N = 2^k - 1$ and $q_{\mathsf{tag}} = 2^\ell - 1$ be positive integers, and $B_{\mathbf{s}} > 0$ be a norm bound. Then there exists a one-time simulation-sound NIZK argument system for the language $\mathcal{L}^{\mathsf{cca}} = (\mathcal{L}^{\mathsf{cca}}_{\mathrm{zk}}, \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}})$ defined as follows:*

$$
\begin{aligned}
\mathcal{L}^{\mathsf{cca}}_{\mathsf{zk}} = \{ & (\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{2m+k+\ell} \times \mathbb{Z}_q^{2m+k+\ell} : \\
& \exists (\mathbf{s}, \mathsf{id}, \tau_{\mathsf{id}}, \{(\mathbf{r}_i, \mathbf{e}_i)\}_{i \in \{1,2\}}) \in \mathbb{Z}^m \times [N] \times [q_{\mathsf{tag}}] \times \left(\mathbb{Z}_q^n \times \mathbb{Z}^{2m+k+\ell}\right)^2 \ \ s.t. \\
& \quad \mathbf{t} = \mathbf{A} \cdot \mathbf{s} + \mathsf{id} \cdot \mathbf{b} + \tau_{\mathsf{id}} \cdot \mathbf{d} \bmod q \\
& \quad \wedge \forall i \in [2] : \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \end{pmatrix} \bmod q \\
& \quad \wedge \|\mathbf{s}\|_2 \leq B_{\mathbf{s}} \wedge \|\mathbf{e}_1\|_2 \leq B_{\mathbf{e}} \wedge \|\mathbf{e}_2\|_2 \leq B_{\mathbf{e}} \}, \qquad (22)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}} = \{ & (\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{2m+k+\ell} \times \mathbb{Z}_q^{2m+k+\ell} : \\
& \exists (\overline{c}, (\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}), \overline{\mathsf{id}}, \overline{\tau}_{\mathsf{id}}, \{(\overline{\mathbf{r}}_i, \overline{\mathbf{e}}_i)\}_{i \in \{1,2\}}) \in [C] \times \mathbb{Z}^{m+k+\ell} \times \mathbb{Z}^2 \times \left(\mathbb{Z}_q^n \times \mathbb{Z}^{2m+k+\ell}\right)^2 \ \ s.t. \\
& \quad \overline{c} \cdot \mathbf{t} = \mathbf{A} \cdot \overline{\mathbf{s}} + \overline{\mathsf{id}} \cdot \mathbf{b} + \overline{\tau}_{\mathsf{id}} \cdot \mathbf{d} \bmod q \\
& \quad \wedge \forall i \in [2] : \overline{c} \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}) \end{pmatrix} \bmod q \\
& \quad \wedge \|(\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}})\|_2 \leq 2\alpha\sqrt{m+k+\ell} \\
& \quad \wedge \|\overline{\mathbf{e}}_1\|_2 \leq 2\alpha\sqrt{2m+k+\ell} \wedge \|\overline{\mathbf{e}}_2\|_2 \leq 2\alpha\sqrt{2m+k+\ell} \\
& \quad \wedge \overline{\mathsf{id}} = \left(1 \mid 2 \mid \cdots \mid 2^{k-1}\right) \cdot \overline{\mathbf{s}}_{\mathsf{id}} \quad \wedge \overline{\tau}_{\mathsf{id}} = \left(1 \mid 2 \mid \cdots \mid 2^{\ell-1}\right) \cdot \overline{\mathbf{s}}_{\tau_{\mathsf{id}}} \}, \qquad (23)
\end{aligned}
$$

where for $i \in \{1,2\}$, $(\mathbf{U}_i, \mathbf{V}_i) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times (m+k+\ell)}$ defines a public key of dual-Regev encryption scheme. The parameters $B_{\mathbf{e}}, C, \alpha, K, \gamma \geq 1$ and the number of parallel executions κ satisfy

$$\alpha = \gamma \cdot \sqrt{\kappa} \cdot C \cdot \sqrt{2B_{\mathbf{e}}^2 + B_{\mathbf{s}}^2 + k + \ell},$$

and

$$
\begin{gathered}
\alpha\sqrt{m+k+\ell} < K/2; \quad \alpha\sqrt{2m+k+\ell} < q/4K; \\
\alpha\sqrt{2m+k+\ell} \cdot \sqrt{1 + n\lceil \log q \rceil (m - n\lceil \log q \rceil)} < q/2\sqrt{5}.
\end{gathered}
$$

Proof. We shows that a trapdoor Σ-protocol for $\mathcal{L}^{\mathsf{cca}}$ exists, an NIZK argument system is then obtained by applying the compiler of [45].

We apply the trapdoor Σ-protocol in Section 3.2. Namely, we execute the algorithm $\mathsf{Gen}_{\mathcal{L}}$ on input public parameters $\mathsf{par} = \{q, n, m, l = m+k+\ell\}$. Next, let the language-specific information $\mathsf{info}_{\mathcal{L}}$ be the matrix $\mathbf{A}$ and the ℓ_2-norm bound $B = \sqrt{B_{\mathbf{s}}^2 + k + \ell}$, we run algorithm $\mathsf{Gen}_{\mathcal{L}}$ to obtain

$$\mathsf{crs}_{\mathcal{L}^{\mathsf{cca}}} = \left(\mathbf{A}, \{(\mathbf{U}_i, \mathbf{V}_i)\}_{i \in [2]}, B, \chi, B_{\mathbf{e}}, C, \alpha, K\right).$$

Note that conditions (5) (with the width adjusted to κ parallel executions as discussed above) and (6) impose the following constraints:

$$\alpha = \gamma \cdot \sqrt{\kappa} \cdot C \cdot \sqrt{2B_{\mathbf{e}}^2 + B_{\mathbf{s}}^2 + k + \ell},$$

and

$$\alpha\sqrt{m+k+\ell} < K/2, \quad \alpha \cdot \sqrt{2m+k+\ell} < q/4K,$$

and

$$\alpha\sqrt{2m+k+\ell} \cdot \sqrt{1 + n\lceil \log q \rceil (m - n\lceil \log q \rceil)} < q/2\sqrt{5}.$$

The string $\mathsf{crs}_{\mathcal{L}^{\mathsf{cca}}}$ fully determines the language $\mathcal{L}^{\mathsf{cca}} = (\mathcal{L}^{\mathsf{cca}}_{\mathrm{zk}}, \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}})$. To prove a statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathcal{L}^{\mathsf{cca}}_{\mathrm{zk}}$, we rewrite the equation

$$\mathbf{t} = \mathbf{A} \cdot \mathbf{s} + \mathsf{id} \cdot \mathbf{b} + \tau_{\mathsf{id}} \cdot \mathbf{d} \bmod q,$$

as $\mathbf{t} = \mathbf{R} \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \bmod q$ where

$$\mathbf{R} = \left(\mathbf{A} \mid \left(\mathbf{b} \mid 2 \cdot \mathbf{b} \mid \cdots \mid 2^{k-1} \cdot \mathbf{b}\right) \mid \left(\mathbf{d} \mid 2 \cdot \mathbf{d} \mid \cdots \mid 2^{\ell-1} \cdot \mathbf{d}\right)\right) \in \mathbb{Z}_q^{n \times (m+k+\ell)}.$$

Then run the prover of the Σ-protocol in Section 3.2 with the transformed statement $(\mathbf{R}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ and witnesses $\mathbf{x} = (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \in \mathbb{Z}^{m+k+\ell}; \mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^{2m+k+\ell}$ and $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}_q^n$ satisfying (21).

From the definition of $\mathcal{L}_{\mathrm{snd}}$ in (9), soundness is guaranteed for statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ satisfying

$$\begin{aligned} \overline{c} \cdot \mathbf{t} &= \mathbf{R} \cdot \overline{\mathbf{x}} \bmod q, \\ \forall i \in [2] : \overline{c} \cdot \mathbf{c}_i &= \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot \overline{\mathbf{x}} \end{pmatrix} \bmod q. \end{aligned}$$

where $\overline{c} \in [C]$, $\|\overline{\mathbf{x}}\|_2 \leq 2\alpha\sqrt{m+k+\ell}$ and $\|\overline{\mathbf{e}}_1\|_2, \|\overline{\mathbf{e}}_2\|_2 \leq 2\alpha\sqrt{2m+k+\ell}$ and

$$\mathbf{R} = \left(\mathbf{A} \mid \left(\mathbf{b} \mid 2 \cdot \mathbf{b} \mid \cdots \mid 2^{k-1} \cdot \mathbf{b}\right) \mid \left(\mathbf{d} \mid 2 \cdot \mathbf{d} \mid \cdots \mid 2^{\ell-1} \cdot \mathbf{d}\right)\right) \in \mathbb{Z}_q^{n \times (m+k+\ell)}.$$

By parsing $\overline{\mathbf{x}} = (\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}) \in \mathbb{Z}^m \times \mathbb{Z}^k \times \mathbb{Z}^\ell$ and rewriting two equations above, we obtain

$$\begin{aligned} \overline{c} \cdot \mathbf{t} &= \mathbf{A} \cdot \overline{\mathbf{s}} + \overline{\mathsf{id}} \cdot \mathbf{b} + \overline{\tau}_{\mathsf{id}} \cdot \mathbf{d} \bmod q, \\ \forall i \in [2] : \overline{c} \cdot \mathbf{c}_i &= \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}) \end{pmatrix} \bmod q, \end{aligned}$$

where $\overline{\mathsf{id}} = (1 \mid 2 \mid \cdots \mid 2^{k-1}) \cdot \overline{\mathbf{s}}_{\mathsf{id}}$ and $\overline{\tau}_{\mathsf{id}} = (1 \mid 2 \mid \cdots \mid 2^{\ell-1}) \cdot \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}$. This shows that $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}}$ as defined in (23). □

We remark that the above NIZK argument system only guarantees that $\overline{c} \cdot \mathbf{c}_1$ and $\overline{c} \cdot \mathbf{c}_2$ are well-formed dual-Regev ciphertexts for some small scalar $\overline{c}$. This serves as an important technical point for arguing traceability, as the reduction needs to decrypt the multiple $\overline{c} \cdot \mathbf{c}_1$ to extract an SIS solution. Additionally, as an adversary against CCA-full anonymity can open signatures of its choice, the parameters must be set so that the technique of switching decryption key [60] does not lead to inconsistency in the output of opening oracle.

4.2 The Construction

Let λ be the security parameter and let $n = \mathcal{O}(\lambda)$ be a lattice dimension. The scheme works with group size $N = 2^k - 1 \in \mathsf{poly}(\lambda)$ and depends on parameters and distributions described below:

- A sufficiently large prime modulus $q \in \mathsf{poly}(\lambda)$, we require that $N < q/2$;
- A dimension $m = 2n\lceil \log q \rceil + \Omega(\lambda)$;
- A small Gaussian width $\sigma \geq \mathcal{O}(1) \cdot \sqrt{1+m^2}$, this parameter dictates the length of user's secret key;
- A Gaussian width $\sigma_{\mathsf{sign}} \geq \max\left(\mathcal{O}(1) \cdot \sqrt{1+2(m\sigma)^2}, \sqrt{m} \cdot \ell_{\mathbf{m}}\right)$;
- A norm bound $B_{\mathbf{s}} = \sigma_{\mathsf{sign}} \cdot \sqrt{3m}$;
- A positive integer $q_{\mathsf{tag}} = \mathsf{poly}(\lambda)$ specifying the tag space in JRS signature scheme [36]. We require that $q_{\mathsf{tag}} = 2^\ell - 1$ for some $\ell \in \mathbb{Z}$ and that $q_{\mathsf{tag}} < q/2$;
- A integer $\ell_{\mathbf{m}} = \mathsf{poly}(\lambda)$ specifying the maximum bit-length of messages;
- A distribution χ over $\mathbb{Z}$, we require that there exists $B_{\mathbf{e}} = \mathsf{poly}(\lambda)$ such that elements sampled from $\chi^{2m+k+\ell}$ have ℓ_2-norm bounded by $B_{\mathbf{e}}$ with overwhelming probability. The distribution χ is used for sampling randomness in dual-Regev encryption;
- A positive integer $C = \mathsf{poly}(\lambda)$ defining the challenge space of the trapdoor Σ-protocol presented in Section 3.2;
- A parameter $\gamma \geq 1$, a width $\alpha > 0$ defining the distribution of responses in the trapdoor Σ-protocol presented in Section 3.2, and the rejection rate $M = \exp(12/\gamma + 1/(2\gamma^2))$ for the κ parallel executions of the protocol;
- An integer $K = \mathsf{poly}(\lambda)$ that defines the plaintext space of the dual-Regev encryption scheme;
- An integer $\kappa = \Theta(\lambda/\log(\lambda))$, which is the number of repetitions of the trapdoor Σ-protocol.

We remark that χ, C, α and κ are chosen by setting up the common reference string of the NIZK argument system for the language $\mathcal{L}^{\mathsf{cca}}$. By Proposition 4.1, these parameters should satisfy:

$$\alpha = \gamma \cdot \sqrt{\kappa} \cdot C \cdot \sqrt{2B_{\mathbf{e}}^2 + B_{\mathbf{s}}^2 + k + \ell} = \gamma \cdot \sqrt{\kappa} \cdot C \cdot \sqrt{2B_{\mathbf{e}}^2 + 3\sigma_{\mathsf{sign}}^2 m + k + \ell}, \quad (24)$$

and

$$\alpha\sqrt{m+k+\ell} < K/2; \qquad \alpha \cdot \sqrt{2m+k+\ell} < q/4K,$$
$$\alpha \cdot \sqrt{2m+k+\ell} \cdot \sqrt{1 + n\lceil \log q \rceil (m - n\lceil \log q \rceil)} < q/2\sqrt{5}.$$

For the correctness of the opening algorithm and for the anonymity proof, let $\beta_{\mathsf{dec}} = 2\alpha\sqrt{2m+k+\ell} \cdot \sqrt{m+1}$. We require that

$$4C\alpha\sqrt{m+k+\ell} < K/2, \quad 2C \cdot \beta_{\mathsf{dec}} < \lfloor q/2K \rfloor. \quad (25)$$

The parameters of the NIZK argument system are set up so that (24) and (25) are met.

We describe the algorithms underlying our CCA-secure group signature:

– $\mathsf{KeyGen}(1^\lambda, 1^N)$: Given a security parameter λ and the desired number of group members $N = 2^k - 1 \in \mathsf{poly}(\lambda)$, which defines the identity space $\mathcal{ID} = [N]$, the algorithm chooses parameters and does the following:
  1. Sample $\mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$, $\mathbf{T} \leftarrow \mathcal{U}(\{0,1\}^{m\times m})$, $\mathbf{C} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{n\times m}\right)$. Let $\mathbf{B} = \mathbf{AT} \bmod q \in \mathbb{Z}_q^{n\times m}$;
  2. Choose a positive integer $\ell_{\mathbf{m}} = \mathsf{poly}(\lambda)$ specifying the message space $\{0,1\}^{\ell_{\mathbf{m}}}$, then sample a matrix $\mathbf{D} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times \ell_{\mathbf{m}}})$;
  3. For each user $\mathsf{id} \in [N]$, generate signing key for id by forming the matrix

$$\mathbf{A}_{\mathsf{id}} = (\mathbf{A} | \mathsf{id} \cdot \mathbf{G}_{n,m} - \mathbf{B}) \in \mathbb{Z}_q^{n\times 2m}$$

  and sampling a matrix $\mathbf{R}_{\mathsf{id}} \in \mathbb{Z}^{2m\times m}$ so that $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{R}_{\mathsf{id}} = \mathbf{C} \bmod q$ and the columns of $\mathbf{R}_{\mathsf{id}}$ are independently distributed as $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}_{\mathsf{id}}),\sigma}$. This is done by running algorithm $\mathsf{SampleD}$ of Lemma 2.4 with $\mathbf{T}$ as a $\mathbf{G}$-trapdoor of $\mathbf{A}_{\mathsf{id}}$ with tag $\mathsf{id}\cdot\mathbf{I}_n$, which is possible as $\|\mathbf{T}\|_2 \le m$. Then set $\mathsf{gsk}[\mathsf{id}] = \mathbf{R}_{\mathsf{id}}$. We remark that for any $\tau \in [q_{\mathsf{tag}}]$, $(\mathbf{A}_{\mathsf{id}} \,|\, \tau \cdot \mathbf{G}_{n,m} - \mathbf{C})$ admits $\mathbf{R}_{\mathsf{id}}$ as a $\mathbf{G}$-trapdoor with tag $\tau \cdot \mathbf{I}_n$. Moreover, each column of $\mathbf{R}_{\mathsf{id}}$ has norm at most $\sigma\sqrt{2m}$ except with negligible probability (Lemma 2.1), and thus $\|\mathbf{R}_{\mathsf{id}}\|_2 \le \sqrt{2} \cdot m\sigma$.
  4. Generate a common reference string crs of the simulation-sound NIZK argument system of [45] (see Appendix B.3) as follows:
     - Run algorithm $\mathsf{Gen}_{\mathcal{L}}$ of the trapdoor Σ-protocol in Section 3.2; using public parameters $(q, n, m, l = m+k+\ell)$ and language-specific norm bound $B = \sqrt{B_{\mathbf{s}}^2 + k + \ell} = \sqrt{3\sigma_{\mathsf{sign}}^2 m + k + \ell}$. Let

$$\mathsf{crs}_{\mathcal{L}} = \left(\{\mathbf{U}_i, \mathbf{V}_i\}_{i\in\{1,2\}}, B, \chi, B_{\mathbf{e}}, C, \alpha, K\right)$$

       be the output and let $\mathbf{S}_1, \mathbf{S}_2 \in \{0,1\}^{m\times(m+k+\ell)}$ be the two dual-Regev decryption keys generated during the execution of $\mathsf{Gen}_{\mathcal{L}}$. Set $\mathsf{gmsk} = \mathbf{S}_1$;
     - Choose a strongly unforgeable one-time signature scheme $\Pi^{\mathsf{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell_0 \in \mathrm{poly}(\lambda)$;
     - Choose a $\mathcal{R}_{\mathsf{BM}}$-lossy encryption scheme

$$\mathsf{RLPKE} = (\mathsf{Param}, \mathsf{KeyGen}, \mathsf{LKeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \mathsf{LOpen}),$$

       for relation $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ defined by an admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_0} \to \{0,1\}^L$. We assume that the space $\mathbb{Z}_q^n \times (\mathbb{Z}_q^{2m+k+\ell})^2$ of first-move messages in the trapdoor Σ-protocol of Section 3.2 can be embedded to the message space of the $\mathcal{R}_{\mathsf{BM}}$-lossy encryption scheme.
       The algorithm then generates public parameters $\Gamma \leftarrow \mathsf{Param}(1^\lambda)$, chooses a random initialization value $K \leftarrow \mathcal{K}$ and generate lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)$.
     - Generate a key $k \leftarrow \mathrm{Gen}(1^\lambda)$ of a somewhere CI hash function family.
     - Choose a number of parallel repetitions $\kappa = \Theta(\lambda/\log(\lambda))$.

Let

$$\mathsf{gpk} = (\mathbf{u}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \underbrace{\underbrace{\mathbf{A}, \{\mathbf{U}_i, \mathbf{V}_i\}_{i\in\{1,2\}}, B, \chi, B_{\mathbf{e}}, C, \alpha, K}_{\mathsf{crs}_{\mathcal{L}^{\mathsf{cca}}}}, \mathsf{AHF}, \Pi^{\mathsf{ots}}, \Gamma, pk, k, \kappa)}_{\mathsf{crs}}.$$

The algorithm outputs $(\mathsf{gpk}, \mathsf{gmsk}, \{\mathsf{gsk}[\mathsf{id}]\}_{\mathsf{id}\in[N]})$.

– $\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[\mathsf{id}], \mathbf{m})$: A user id initializes a private state $\tau_{\mathsf{id}} \leftarrow 0$. To sign a message $\mathbf{m} \in \{0,1\}^{\ell_{\mathbf{m}}}$, the user retrieves $\mathsf{gsk}[\mathsf{id}] = \mathbf{R}_{\mathsf{id}}$ and does the following:
  1. Check if $\tau_{\mathsf{id}} < q_{\mathsf{tag}}$, return $\perp$ if it is not the case;
  2. Set $\tau_{\mathsf{id}} \leftarrow \tau_{\mathsf{id}} + 1$, then create a JRS signature with tag τ_{id}. This is done by sampling a vector $\mathbf{x}$ from the distribution $\mathcal{D}^{3m}_{\sigma_{\mathsf{sign}}}$ conditioned on

$$(\mathbf{A}_{\mathsf{id}} \,|\, \tau_{\mathsf{id}} \cdot \mathbf{G} - \mathbf{C}) \cdot \mathbf{x} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} \bmod q.$$

     The vector $\mathbf{x}$ is sampled by executing algorithm $\mathsf{SampleD}$ of Lemma 2.4, using the $\mathbf{G}$-trapdoor of $(\mathbf{A}_{\mathsf{id}} \,|\, \tau_{\mathsf{id}} \cdot \mathbf{G} - \mathbf{C})$ with tag $\tau_{\mathsf{id}} \cdot \mathbf{I}_n$;
  3. Parse $\mathbf{x} = (\mathbf{s}, \mathbf{s}_1, \mathbf{s}_2) \in \mathbb{Z}^m \times \mathbb{Z}^m \times \mathbb{Z}^m$. For $i \in \{1,2\}$, encrypt the message $(\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \in \mathbb{Z}^m \times \{0,1\}^k \times \{0,1\}^\ell$ under dual-Regev encryption scheme by sampling $\mathbf{r}_i \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{e}_i \leftarrow \chi^{2m+k+\ell}$ and outputting a ciphertext

$$\mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \mathbf{r}_i + \mathbf{e}_i + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \end{pmatrix} \bmod q \in \mathbb{Z}_q^{2m+k+\ell}.$$

  4. Compute $\mathbf{t} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} + \mathbf{B} \cdot \mathbf{s}_1 + \mathbf{C} \cdot \mathbf{s}_2 \in \mathbb{Z}_q^n$, $\mathbf{b} = \mathbf{G}_{n,m} \cdot \mathbf{s}_1 \bmod q$ and $\mathbf{d} = \mathbf{G}_{n,m} \cdot \mathbf{s}_2 \bmod q$. Then generate an NIZK argument π for the statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ that satisfies

$$\begin{aligned} \mathbf{t} &= \mathbf{A} \cdot \mathbf{s} + \mathsf{id} \cdot \mathbf{b} + \tau_{\mathsf{id}} \cdot \mathbf{d} \bmod q; \\ \mathbf{c}_1 &= \begin{pmatrix} \mathbf{U}_1^\top \\ \mathbf{V}_1^\top \end{pmatrix} \cdot \mathbf{r}_1 + \mathbf{e}_1 + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \end{pmatrix} \bmod q, \\ \mathbf{c}_2 &= \begin{pmatrix} \mathbf{U}_2^\top \\ \mathbf{V}_2^\top \end{pmatrix} \cdot \mathbf{r}_2 + \mathbf{e}_2 + \begin{pmatrix} \mathbf{0}^m \\ \lfloor q/2K \rfloor \cdot (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \end{pmatrix} \bmod q \end{aligned} \tag{26}$$

     using witness $(\mathbf{s}, \mathsf{id}, \tau_{\mathsf{id}}, \mathbf{r}_1, \mathbf{e}_1, \mathbf{r}_2, \mathbf{e}_2)$.

  Output the final signature

$$\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)\,. \tag{27}$$

– $\mathsf{Verify}(\mathsf{gpk}, \mathbf{m}, \Sigma)$: parse the signature Σ as in (27). Compute $\mathbf{t} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} + \mathbf{B} \cdot \mathbf{s}_1 + \mathbf{C} \cdot \mathbf{s}_2 \bmod q \in \mathbb{Z}_q^n$, $\mathbf{b} = \mathbf{G}_{n,m} \cdot \mathbf{s}_1 \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{d} = \mathbf{G}_{n,m} \cdot \mathbf{s}_2 \bmod q \in \mathbb{Z}_q^n$. Output 1 if and only if the following conditions are satisfied:
  - $\|(\mathbf{s}_1, \mathbf{s}_2)\|_2 \leq \sigma_{\mathsf{sign}} \sqrt{2m}$;

- π is a valid NIZK argument for statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ of language $\mathcal{L}^{\mathsf{cca}}$ (defined in Proposition 4.1).

– $\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{m}, \Sigma)$: return $\perp$ if $\mathsf{Verify}(\mathsf{gpk}, \mathbf{m}, \Sigma) \neq 1$. Otherwise, parse Σ as in (27) and compute $\mathbf{t} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} + \mathbf{B} \cdot \mathbf{s}_1 + \mathbf{C} \cdot \mathbf{s}_2 \in \mathbb{Z}_q^n$, $\mathbf{b} = \mathbf{G}_{n,m} \cdot \mathbf{s}_1 \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{d} = \mathbf{G}_{n,m} \cdot \mathbf{s}_2 \bmod q \in \mathbb{Z}_q^n$. Let $\Delta = \lfloor q/2K \rfloor$. Then for each $c \in [C]$, use $\mathsf{gmsk} = \mathbf{S}_1 \in \{0,1\}^{m \times (m+k+\ell)}$ to perform the below steps:

(I) Compute $\mathbf{v} = \left(-\mathbf{S}_1^\top \mid \mathbf{I}_{m+k+\ell}\right) \cdot (c \cdot \mathbf{c}_1) \bmod q \in \mathbb{Z}_q^{m+k+\ell}$.

(II) For each $h \in [m+k+\ell]$, find a pair $(\mathbf{p}[h], \mathbf{y}[h]) \in \mathbb{Z}^2$ such that $(\mathbf{v}[h] - \Delta \cdot \mathbf{p}[h]) \bmod q = \mathbf{y}[h]$, $|\mathbf{p}[h]| < K/2$ and $|\mathbf{y}[h]| \leq \beta_{\mathsf{dec}}$, where the reduction modulo q returns the representative in $(-q/2, q/2)$. If such a pair does not exist for some h, restart from (I) with the next c. Otherwise, let $\mathbf{p} = (\mathbf{p}[h])_{h \in [m+k+\ell]}$.

(III) Parse $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \in \mathbb{Z}^m \times \mathbb{Z}^k \times \mathbb{Z}^\ell$, and let $a_{\mathsf{id}} = \sum_{i \in [k]} \mathbf{p}_2[i] \cdot 2^{i-1} \in \mathbb{Z}$ and $a_\tau = \sum_{i \in [\ell]} \mathbf{p}_3[i] \cdot 2^{i-1} \in \mathbb{Z}$. Check whether $\|\mathbf{p}\|_2 \leq 2\alpha\sqrt{m+k+\ell}$ and

$$\mathbf{A} \cdot \mathbf{p}_1 + a_{\mathsf{id}} \cdot \mathbf{b} + a_\tau \cdot \mathbf{d} = c \cdot \mathbf{t} \bmod q.$$

If any check fails, restart from (I) with the next c.

(IV) If there exists an integer $\mathsf{id} \in [N]$ such that $a_{\mathsf{id}} = c \cdot \mathsf{id} \bmod q$, return id. Otherwise, restart from (I) with the next c.

In case the above steps do not output a valid id with any $c \in [C]$, return $\perp$.

For $i \in \{1, 2\}$, we denote by $\mathsf{Open}^{(i)}$ the algorithm Open as above, except that Step (I) is performed with respect to the decryption key $\mathbf{S}_i$ and the ciphertext $\mathbf{c}_i$. By definition, $\mathsf{Open}^{(1)} = \mathsf{Open}$.

We remark that the pair found in Step (II) is unique for each h, if it exists. Indeed, let $w \in \mathbb{Z}_q$ and assume that $(a, y_a), (b, y_b) \in \mathbb{Z}^2$ are such that $(w - \Delta a) \bmod q = y_a$, $(w - \Delta b) \bmod q = y_b$, $|a|, |b| < K/2$ and $|y_a|, |y_b| \leq \beta_{\mathsf{dec}}$. Then $\Delta \cdot (b - a) = y_a - y_b \bmod q$, where $|\Delta \cdot (b - a)| < K \cdot \Delta \leq q/2$ and $|y_a - y_b| \leq 2\beta_{\mathsf{dec}} < \Delta \leq q/2$ by (25). Hence $\Delta \cdot (b - a) = y_a - y_b$ over the integers, so that $|\Delta \cdot (b - a)| < \Delta$, and therefore $a = b$ and $y_a = y_b$.

4.3 Security Analysis

Theorem 4.1. *The above group signature scheme is correct assuming that the NIZK argument system are correct.*

Proof. Let $\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)$ be an honestly generated signature on a message $\mathbf{m} \in \{0,1\}^{\ell_{\mathbf{m}}}$ by the user $\mathsf{id} \in [N]$. Let $\mathbf{x} = (\mathbf{s}, \mathbf{s}_1, \mathbf{s}_2)$ and $\tau_{\mathsf{id}} \in [q_{\mathsf{tag}}]$ be the JRS signature and the tag computed in Sign, and let $\mathbf{x}' = (\mathbf{s}, \mathsf{bin}(\mathsf{id}), \mathsf{bin}(\tau_{\mathsf{id}})) \in \mathbb{Z}^{m+k+\ell}$. Let $\Delta = \lfloor q/2K \rfloor$. We first show that Verify outputs 1 except with a negligible probability. This follows from:

– The bound on $(\mathbf{s}_1, \mathbf{s}_2)$. The matrix $(\mathbf{A} \mid \mathsf{id} \cdot \mathbf{G}_{n,m} - \mathbf{B} \mid \tau_{\mathsf{id}} \cdot \mathbf{G}_{n,m} - \mathbf{C})$ is statistically close to uniform by Lemma 2.2. Thus Lemma 2.5, applied with $\mathbf{A}_1 = \mathbf{A}$ and $\mathbf{A}_2 = (\mathsf{id} \cdot \mathbf{G}_{n,m} - \mathbf{B} \mid \tau_{\mathsf{id}} \cdot \mathbf{G}_{n,m} - \mathbf{C})$, shows that $(\mathbf{s}_1, \mathbf{s}_2)$

is statistically close to $\mathcal{D}^{2m}_{\sigma_{\mathsf{sign}}}$. By Lemma 2.1, we have that $\|(\mathbf{s}_1, \mathbf{s}_2)\|_2 \leq \sigma_{\mathsf{sign}}\sqrt{2m}$ except with negligible probability.

– The correctness of the underlying NIZK system. By Lemma 2.1, we have $\|\mathbf{s}\|_2 \leq \|\mathbf{x}\|_2 \leq \sigma_{\mathsf{sign}}\sqrt{3m} = B_{\mathbf{s}}$, and $\|\mathbf{e}_1\|_2, \|\mathbf{e}_2\|_2 \leq B_{\mathbf{e}}$ except with negligible probability. Moreover, as $(\mathbf{A} \mid \mathsf{id} \cdot \mathbf{G}_{n,m} - \mathbf{B} \mid \tau_{\mathsf{id}} \cdot \mathbf{G}_{n,m} - \mathbf{C}) \cdot \mathbf{x} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} \bmod q$, we have that $\mathbf{t} = \mathbf{A}\cdot\mathbf{s} + \mathsf{id}\cdot\mathbf{b} + \tau_{\mathsf{id}}\cdot\mathbf{d} \bmod q$. Therefore the statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ belongs to $\mathcal{L}^{\mathsf{cca}}_{\mathsf{zk}}$ (22).

For the correctness of Open, consider the first iteration of Open, namely $c = 1$. We have that

$$\mathbf{c}_1 = \begin{pmatrix} \mathbf{U}_1^\top \\ \mathbf{V}_1^\top \end{pmatrix} \cdot \mathbf{r}_1 + \mathbf{e}_1 + \begin{pmatrix} \mathbf{0}^m \\ \Delta \cdot \mathbf{x}' \end{pmatrix} \bmod q,$$

where $\|\mathbf{e}_1\|_2 \leq B_{\mathbf{e}}$. By (24), we have that $\alpha \geq \sqrt{B_{\mathbf{s}}^2 + k + \ell}$ and $\alpha \geq B_{\mathbf{e}}$, and therefore $\|\mathbf{x}'\|_2 \leq \alpha < 2\alpha\sqrt{m + k + \ell}$ and $\|\mathbf{e}_1\|_2 \leq \alpha < 2\alpha\sqrt{2m + k + \ell}$. Since $\mathbf{V}_1 = \mathbf{U}_1\mathbf{S}_1 \bmod q$, Step (I) computes

$$\mathbf{v} = \left(-\mathbf{S}_1^\top \mid \mathbf{I}_{m+k+\ell}\right) \cdot \mathbf{c}_1 = \Delta \cdot \mathbf{x}' + \mathbf{y}_1 \bmod q, \quad \mathbf{y}_1 = \left(-\mathbf{S}_1^\top \mid \mathbf{I}_{m+k+\ell}\right) \cdot \mathbf{e}_1.$$

Each row of $\left(-\mathbf{S}_1^\top \mid \mathbf{I}_{m+k+\ell}\right)$ has norm at most $\sqrt{m+1}$, thus $\|\mathbf{y}_1\|_\infty \leq \sqrt{m+1} \cdot \|\mathbf{e}_1\|_2 \leq \beta_{\mathsf{dec}}$. Moreover, $\|\mathbf{x}'\|_\infty \leq 2\alpha\sqrt{m + k + \ell} < K/2$ by (25). Hence, for each $h \in [m + k + \ell]$, the pair $(\mathbf{x}'[h], \mathbf{y}_1[h])$ satisfies the requirement of Step (II), and it is the unique such pair. Thus Step (II) recovers $\mathbf{p} = \mathbf{x}'$. Then all checks of Step (III) pass, as $a_{\mathsf{id}} = \mathsf{id}$ and $a_\tau = \tau_{\mathsf{id}}$. Finally, in Step (IV) we have $a_{\mathsf{id}} = 1 \cdot \mathsf{id}$, and thus Open returns id. □

For the anonymity proof, we need the following intermediate result.

Proposition 4.2. *Let $\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)$ be a signature on $\mathbf{m} \in \{0,1\}^{\ell_{\mathbf{m}}}$, $\mathbf{t} = \mathbf{u} + \mathbf{D} \cdot \mathbf{m} + \mathbf{B} \cdot \mathbf{s}_1 + \mathbf{C} \cdot \mathbf{s}_2 \bmod q \in \mathbb{Z}_q^n$, $\mathbf{b} = \mathbf{G}_{n,m} \cdot \mathbf{s}_1 \bmod q \in \mathbb{Z}_q^n$ and $\mathbf{d} = \mathbf{G}_{n,m} \cdot \mathbf{s}_2 \bmod q \in \mathbb{Z}_q^n$. Assume that the statement $(\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2) \in \mathcal{L}^{\mathsf{cca}}_{\mathsf{snd}}$ (defined in Proposition 4.1). Then the algorithms $\mathsf{Open}^{(1)}$ and $\mathsf{Open}^{(2)}$ have the same output on $(\mathbf{m}, \Sigma)$.*

Essentially, under the conditions as in (25), decrypting $\overline{c} \cdot \mathbf{c}_1$ with $\mathbf{S}_1$ and $\overline{c} \cdot \mathbf{c}_2$ with $\mathbf{S}_2$ yields the same plaintext $\overline{\mathbf{x}}$, where $(\overline{c}, \overline{\mathbf{x}})$ is part of a witness for $\mathcal{L}^{\mathsf{cca}}_{\mathsf{snd}}$. This forces the consistency between $\mathsf{Open}^{(1)}$ and $\mathsf{Open}^{(2)}$.

Proof. Let $\Delta = \lfloor q/2K \rfloor$. By the definition of $\mathcal{L}^{\mathsf{cca}}_{\mathsf{snd}}$ in Proposition 4.1, there exist $\overline{c} \in [C]$, $\overline{\mathbf{x}} = (\overline{\mathbf{s}}, \overline{\mathbf{s}}_{\mathsf{id}}, \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}) \in \mathbb{Z}^{m+k+\ell}$, $\overline{\mathsf{id}}, \overline{\tau}_{\mathsf{id}} \in \mathbb{Z}$ and $(\overline{\mathbf{r}}_i, \overline{\mathbf{e}}_i)_{i\in\{1,2\}}$ such that $\|\overline{\mathbf{x}}\|_2 \leq 2\alpha\sqrt{m + k + \ell}$, $\|\overline{\mathbf{e}}_i\|_2 \leq 2\alpha\sqrt{2m + k + \ell}$ for $i \in \{1, 2\}$, $\overline{\mathsf{id}} = \sum_{i\in[k]} \overline{\mathbf{s}}_{\mathsf{id}}[i] \cdot 2^{i-1}$, $\overline{\tau}_{\mathsf{id}} = \sum_{i\in[\ell]} \overline{\mathbf{s}}_{\tau_{\mathsf{id}}}[i] \cdot 2^{i-1}$, and

$$\overline{c} \cdot \mathbf{t} = \mathbf{A} \cdot \overline{\mathbf{s}} + \overline{\mathsf{id}} \cdot \mathbf{b} + \overline{\tau}_{\mathsf{id}} \cdot \mathbf{d} \bmod q, \tag{28}$$

$$\forall i \in \{1, 2\} : \overline{c} \cdot \mathbf{c}_i = \begin{pmatrix} \mathbf{U}_i^\top \\ \mathbf{V}_i^\top \end{pmatrix} \cdot \overline{\mathbf{r}}_i + \overline{\mathbf{e}}_i + \begin{pmatrix} \mathbf{0}^m \\ \Delta \cdot \overline{\mathbf{x}} \end{pmatrix} \bmod q. \tag{29}$$

For $i \in \{1,2\}$, since $\mathbf{V}_i = \mathbf{U}_i\mathbf{S}_i \bmod q$, multiplying (29) by $\left(-\mathbf{S}_i^\top \mid \mathbf{I}_{m+k+\ell}\right)$ gives

$$\left(-\mathbf{S}_i^\top \mid \mathbf{I}_{m+k+\ell}\right)(\overline{c}\cdot\mathbf{c}_i) = \Delta\cdot\overline{\mathbf{x}} + \overline{\mathbf{y}}_i \bmod q, \quad \overline{\mathbf{y}}_i = \left(-\mathbf{S}_i^\top \mid \mathbf{I}_{m+k+\ell}\right)\overline{\mathbf{e}}_i \in \mathbb{Z}^{m+k+\ell}. \tag{30}$$

Each row of $\left(-\mathbf{S}_i^\top \mid \mathbf{I}_{m+k+\ell}\right)$ has ℓ_2-norm at most $\sqrt{m+1}$, hence $\|\overline{\mathbf{y}}_i\|_\infty \le \sqrt{m+1}\cdot\|\overline{\mathbf{e}}_i\|_2 \le \beta_{\mathsf{dec}}$.

Claim 1. Let $i \in \{1,2\}$ and $c \in [C]$. If the iteration of $\mathsf{Open}^{(i)}$ with respect to c passes Step (III) with some $\mathbf{p} \in \mathbb{Z}^{m+k+\ell}$, then $\overline{c}\cdot\mathbf{p} = c\cdot\overline{\mathbf{x}}$.

By Steps (I)–(III), we have that $\left(-\mathbf{S}_i^\top \mid \mathbf{I}_{m+k+\ell}\right)(c\cdot\mathbf{c}_i) = \Delta\cdot\mathbf{p}+\mathbf{y} \bmod q$ for some $\mathbf{y} \in \mathbb{Z}^{m+k+\ell}$ with $\|\mathbf{y}\|_\infty \le \beta_{\mathsf{dec}}$, and $\|\mathbf{p}\|_2 \le 2\alpha\sqrt{m+k+\ell}$. Multiplying this equation by $\overline{c}$ and using (30) multiplied by c, we obtain $\overline{c}\cdot(\Delta\cdot\mathbf{p}+\mathbf{y}) = c\cdot(\Delta\cdot\overline{\mathbf{x}}+\overline{\mathbf{y}}_i) \bmod q$. Let

$$\mathbf{z} = \overline{c}\cdot\mathbf{p} - c\cdot\overline{\mathbf{x}}, \quad \mathbf{y}' = c\cdot\overline{\mathbf{y}}_i - \overline{c}\cdot\mathbf{y},$$

so that $\Delta\cdot\mathbf{z} = \mathbf{y}' \bmod q$. Since $1 \le c, \overline{c} \le C$, by (25) we have that

$$\|\mathbf{z}\|_\infty \le 4C\alpha\sqrt{m+k+\ell} < K/2, \quad \|\mathbf{y}'\|_\infty \le 2C\cdot\beta_{\mathsf{dec}} < \Delta.$$

Hence $\|\Delta\cdot\mathbf{z}\|_\infty < \Delta K/2 \le q/4$ and $\|\mathbf{y}'\|_\infty < \Delta \le q/4$. The vector $\Delta\cdot\mathbf{z}-\mathbf{y}'$ has each entry divisible by q and of absolute value less than $q/2$, hence $\Delta\cdot\mathbf{z} = \mathbf{y}'$ over the integers. This implies that $\|\Delta\cdot\mathbf{z}\|_\infty < \Delta$, and thus $\mathbf{z} = \mathbf{0}$, which proves Claim 1.

Claim 2. For each $i \in \{1,2\}$, the iteration of $\mathsf{Open}^{(i)}$ with respect to $c = \overline{c}$ passes Step (III) with $\overline{\mathbf{x}}$.

By (30), Step (I) of $\mathsf{Open}^{(i)}$ computes $\Delta\cdot\overline{\mathbf{x}}+\overline{\mathbf{y}}_i \bmod q$. For each $h \in [m+k+\ell]$, we have that $|\overline{\mathbf{x}}[h]| \le 2\alpha\sqrt{m+k+\ell} < K/2$ and $|\overline{\mathbf{y}}_i[h]| \le \beta_{\mathsf{dec}}$. Hence the pair $(\overline{\mathbf{x}}[h], \overline{\mathbf{y}}_i[h])$ satisfies the requirement of Step (II) and is therefore the unique such pair, so that Step (II) recovers $\mathbf{p} = \overline{\mathbf{x}}$. The checks of Step (III) then pass by (28) and the bound on $\overline{\mathbf{x}}$. This proves Claim 2.

We now conclude. Since q is prime and $0 < \overline{c} \le C < q$, the integer $\overline{c}$ is invertible modulo q. Let $\widehat{\mathsf{id}} = \overline{c}^{-1}\cdot\overline{\mathsf{id}} \bmod q$. Consider any iteration of $\mathsf{Open}^{(1)}$ or $\mathsf{Open}^{(2)}$, with respect to some $c \in [C]$, that reaches Step (IV) with $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$. By Claim 1, $\overline{c}\cdot\mathbf{p}_2 = c\cdot\overline{\mathbf{s}}_{\mathsf{id}}$, hence $a_{\mathsf{id}} = \sum_{i\in[k]}\mathbf{p}_2[i]\cdot 2^{i-1} = \overline{c}^{-1}\cdot c\cdot\sum_{i\in[k]}\overline{\mathbf{s}}_{\mathsf{id}}[i]\cdot 2^{i-1} = c\cdot\widehat{\mathsf{id}} \bmod q$. As c is invertible modulo q, the check of Step (IV) passes for $\mathsf{id} \in [N]$ if and only if $\mathsf{id} = \widehat{\mathsf{id}} \bmod q$. Since $N < q/2$, at most one $\mathsf{id} \in [N]$ satisfies this. Therefore, any iteration of either algorithm that reaches Step (IV) returns this unique id if it exists. By Claim 2, both $\mathsf{Open}^{(1)}$ and $\mathsf{Open}^{(2)}$ reach Step (IV) in at least one iteration. Hence both algorithms output id if there is $\mathsf{id} \in [N]$ with $\mathsf{id} = \widehat{\mathsf{id}} \bmod q$, and both output $\perp$ otherwise. □

Due to space restriction, the anonymity and unforgeability proofs are given in Appendix C.

References

1. Michel Abdalla and Bogdan Warinschi. On the minimal assumptions of group signature schemes. In *ICICS 2004*, volume 3269 of *LNCS*, pages 1–13. Springer, 2004.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *STOC 1996*, pages 99–108. ACM, 1996.
3. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.
4. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, 2008.
5. Mihir Bellare, Daniele Micciancio, and Bogdan 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.
6. Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In *EUROCRYPT 2022*, volume 13276 of *LNCS*, pages 95–126. Springer, 2022.
7. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
8. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
9. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *CCS 2004*, pages 168–177. ACM, 2004.
10. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. A framework for practical anonymous credentials from lattices. In *CRYPTO 2023*, volume 14082 of *LNCS*, pages 384–417. Springer, 2023.
11. Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. Efficient post-quantum snarks for RSIS and RLWE and their applications to privacy. In *PQCrypto 2020*, volume 12100 of *LNCS*, pages 247–267. Springer, 2020.
12. Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 89–108. Springer, 2011.
13. Pedro Branco, Arka Rai Choudhuri, Nico Döttling, Abhishek Jain, Giulio Malavolta, and Akshayaram Srinivasan. Black-box non-interactive zero knowledge from vector trapdoor hash. In *EUROCRYPT 2025*, volume 15604 of *LNCS*, pages 64–92. Springer, 2025.
14. Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *ACM CCS 2004*, pages 132–145. ACM, 2004.
15. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *SCN 2004*, volume 3352 of *LNCS*, pages 120–133. Springer, 2004.
16. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer, 2005.
17. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

18. Jan Camenisch, Gregory Neven, and Markus Rückert. Fully anonymous attribute tokens from lattices. In *SCN 2012*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012.
19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In *STOC 2019*, pages 1082–1090. ACM, 2019.
20. Ran Canetti, Alex Lombardi, and Daniel Wichs. Non-interactive zero knowledge and correlation intractability from circular-secure FHE. *IACR Cryptol. ePrint Arch.*, page 1248, 2018.
21. David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. *IACR Cryptol. ePrint Arch.*, 2009.
22. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
23. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
24. Michele Ciampi, Roberto Parisella, and Daniele Venturi. On adaptive security of delayed-input sigma protocols and fiat-shamir nizks. In *SCN 2020*, volume 12238 of *LNCS*, pages 670–690. Springer, 2020.
25. Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In *TCC 2006*, volume 3876 of *LNCS*, pages 41–59. Springer, 2006.
26. Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *CCS 2018*, pages 574–591. ACM, 2018.
27. Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *CRYPTO 2014*, volume 8616, pages 335–352. Springer, 2014.
28. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *FOCS 1990*, pages 308–317. IEEE Computer Society, 1990.
29. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
30. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC 2009*, pages 169–178. ACM, 2009.
31. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
32. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.
33. Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS 2010*, pages 230–240, 2010.
34. S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 395–412. Springer, 2010.
35. Tibor Jager. Verifiable random functions from weaker assumptions. In *TCC 2015*, volume 9015 of *LNCS*, pages 121–143. Springer, 2015.

36. Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Lattice signature with efficient protocols, application to anonymous credentials. In *CRYPTO 2023*, volume 14082 of *LNCS*, pages 351–383. Springer, 2023.
37. Shuichi Katsumata and Shota Yamada. Group signatures without NIZK: from lattices in the standard model. In *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 312–344. Springer, 2019.
38. Akinori Kawachi, Keisuke Tanaka, and Keita 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.
39. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Group encryption. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 181–199. Springer, 2007.
40. Sam Kim and David J. Wu. Multi-theorem preprocessing nizks from lattices. In *CRYPTO 2018*, volume 10992 of *LNCS*, pages 733–765. Springer, 2018.
41. Fabien Laguillaumie, Adeline Langlois, Benoît Libert, and Damien Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013.
42. Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC 2014*, volume 8383 of *LNCS*, pages 345–361. Springer, 2014.
43. Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*, volume 10032 of *LNCS*, pages 373–403, 2016.
44. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong 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.
45. Benoît Libert, Khoa Nguyen, Alain Passelègue, and Radu Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In *ASIACRYPT 2020*, volume 12491 of *LNCS*, pages 128–158. Springer, 2020.
46. Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Bifurcated signatures: Folding the accountability vs. anonymity dilemma into a single private signing scheme. In *EUROCRYPT 2021*, volume 12698 of *LNCS*, pages 521–552. Springer, 2021.
47. Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. One-shot fiat-shamir-based NIZK arguments of composite residuosity and logarithmic-size ring signatures in the standard model. In *EUROCRYPT 2022*, volume 13276 of *LNCS*, pages 488–519. Springer, 2022.
48. Benoît Libert, Thomas Peters, and Moti Yung. Group signatures with almost-for-free revocation. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 571–589. Springer, 2012.
49. Benoît Libert, Thomas Peters, and Moti Yung. Scalable group signatures with revocation. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 609–627. Springer, 2012.
50. San Ling, Khoa Nguyen, and Huaxiong Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC 2015*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015.
51. San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *ACNS 2017*, volume 10355 of *LNCS*, pages 293–312. Springer, 2017.

52. San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Constant-size group signatures from lattices. In *PKC 2018*, volume 10770 of *LNCS*, pages 58–88. Springer, 2018.
53. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
54. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.
55. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In *CRYPTO 2022*, volume 13508 of *LNCS*, pages 71–101. Springer, 2022.
56. Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *ASIACRYPT 2021*, volume 13093 of *LNCS*, pages 218–248. Springer, 2021.
57. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
58. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In *CT-RSA 2011*, volume 6558 of *LNCS*, pages 376–392. Springer, 2011.
59. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
60. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC 1990*, pages 427–437. ACM, 1990.
61. Khoa Nguyen, Fuchun Guo, Willy Susilo, and Guomin Yang. Multimodal private signatures. In *CRYPTO 2022*, volume 13508 of *LNCS*, pages 792–822. Springer, 2022.
62. Khoa Nguyen, Partha Sarathi Roy, Willy Susilo, and Yanhong Xu. Bicameral and auditably private signatures. In *ASIACRYPT 2023*, volume 14439 of *LNCS*, pages 313–347. Springer, 2023.
63. Phong Q. Nguyen, Jiang Zhang, and Zhenfeng Zhang. Simpler efficient group signatures from lattices. In *PKC 2015*, volume 9020 of *LNCS*, pages 401–426. Springer, 2015.
64. Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In *CRYPTO 2019*, volume 11692 of *LNCS*, pages 89–114. Springer, 2019.
65. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC 2005*, pages 84–93. ACM, 2005.
66. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS '99*, pages 543–553. IEEE Computer Society, 1999.
67. Brent Waters. A new approach for non-interactive zero-knowledge from learning with errors. In *STOC 2024*, pages 399–410. ACM, 2024.
68. Brent Waters, Hoeteck Wee, and David J. Wu. New techniques for preimage sampling: Improved nizks and more from LWE. In *EUROCRYPT 2025*, volume 15604 of *LNCS*, pages 3–33. Springer, 2025.
69. Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO 2019*, volume 11692 of *LNCS*, pages 147–175. Springer, 2019.

Appendix

A Cryptographic Primitives

A.1 Security Requirement of Group Signatures

Security of a group signature scheme is formalized via the notions of *traceability* and *anonymity*. Traceability ensures that all signatures, even those created by a coalition of users and the group manager, pooling their secret keys together, can be traced to a member of the forging coalition. The adversary is modeled as a PPT algorithm $\mathcal{A}$ in the experiment described in Figure 2. Adversary $\mathcal{A}$ is allowed to see the secret signing key as well as the signature of any user of its choice. The advantage of $\mathcal{A}$ against traceability of GS is defined as

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = \Pr[\mathbf{Exp}^{\mathsf{trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = 1].$$

Definition A.1 (Full traceability [5]). *A group signature scheme* GS *is said to be fully traceable if for all polynomial* $N(\cdot)$ *and all PPT adversaries* $\mathcal{A}$*, its advantage* $\mathsf{Adv}^{trace}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ *is negligible in the security parameter* λ*.*

$(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{Keygen}(1^\lambda, 1^N)$
$\mathsf{st} \leftarrow (\mathsf{gmsk}, \mathsf{gpk})$
$\mathsf{CU} \leftarrow \emptyset$; $K \leftarrow \varepsilon$; $Cont \leftarrow 1$
while $(Cont = 1)$ **do**
  $(Cont, \mathsf{st}, j) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{gsk}[\cdot],\cdot)}(\mathsf{st}, K)$
  if *(*$Cont = 1$*)* **then**
    $\mathsf{CU} \leftarrow \mathsf{CU} \cup \{j\}$; $K \leftarrow \mathsf{gsk}[j]$
$(M^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{gsk}[\cdot],\cdot)}(\mathsf{st}, K)$
if $\mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 0$ **then**
  return 0
if $\mathsf{Open}(\mathsf{gmsk}, M^*, \Sigma^*) = \perp$ **then**
  return 1
if $\exists j^* \in [N] : (\mathsf{Open}(\mathsf{gmsk}, M^*, \Sigma^*) = j^*) \wedge (j^* \notin \mathsf{CU}) \wedge ((j^*, M^*)$ *not queried to* Sign*)* **then**
  return 1
else
  return 0

Fig. 2. Experiment $\mathbf{Exp}^{\mathsf{trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ against traceability of GS

Anonymity requires that an adversary who does not know the group manager secret key cannot recognize the identity of a user given its signature and its secret signing key. More formally, the adversary is involved in the experiment depicted in Figure 3. The advantage of such an adversary $\mathcal{A}$ against a scheme GS with N members is defined as

$$\mathsf{Adv}^{\mathsf{anon}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = |\Pr[\mathbf{Exp}^{\mathsf{anon}-1}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = 1] - \Pr[\mathbf{Exp}^{\mathsf{anon}-0}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = 1]|.$$

Definition A.2 (CPA/CCA-full anonymity [5,8]). *A group signature* GS *is said to be CPA-fully (resp. CCA-fully) anonymous if for all polynomial* $N(\cdot)$ *and all PPT adversaries* $\mathcal{A}$ *(resp. PPT adversaries* $\mathcal{A}$ *with access to an opening oracle except for the challenge signature), its advantage* $\mathsf{Adv}^{\mathsf{anon}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ *is negligible in the security parameter* λ*.*

$(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{Keygen}(1^\lambda, 1^N)$
$(\mathsf{st}, j_0, j_1, M) \leftarrow \mathcal{A}(\mathsf{gpk}, \mathsf{gsk})$
$\Sigma^* \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b], M)$
$b' \leftarrow \mathcal{A}(\mathsf{st}, \Sigma^*)$
$b' \leftarrow \mathcal{A}^{\mathsf{Open}(\cdot)}(\mathsf{st}, \Sigma^*)$
return b'

Fig. 3. Experiment $\mathbf{Exp}^{\mathsf{anon}-\mathsf{b}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ defining CPA-full anonymity (resp. CCA-full anonymity) of GS, excluding the solid box (resp. dashed box)

In CPA-full anonymity, the adversary is not allowed to query an opening oracle. This relaxed model is precisely the one considered in [34], and was firstly introduced in [8]. In constrast, CCA-full anonymity [5] allows the adversary access to an opening oracle that can be called on any signature except the challenge signature Σ^*.

A.2 Non-interactive Zero Knowledge Argument Systems

We recall the formalization of NIZK systems following [45].

Definition A.3. *Let* $\mathcal{R} = (\mathcal{R}_{\mathrm{zk}}, \mathcal{R}_{\mathrm{snd}})$ *be a pair of NP-relation such that* $\mathcal{R}_{\mathrm{zk}} \subseteq \mathcal{R}_{\mathrm{snd}}$*, the associated language be* $\mathcal{L} = (\mathcal{L}_{\mathrm{zk}}, \mathcal{L}_{\mathrm{snd}})$*. An NIZK argument system for* $\mathcal{R}$ *is a tuple* $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim})$ *of PPT algorithms, defined as follows.*

$\mathsf{Setup}(1^\lambda, \mathsf{info}_{\mathcal{L}})$ *taking as input* λ*, a language-specific information* $\mathsf{info}_{\mathcal{L}}$*, outputs a common reference string* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ *and a simulation trapdoor* τ_{sim}*. Here,* $\mathsf{crs}_{\mathcal{L}}$ *is a language reference string giving the full description of* $\mathcal{L}$*.*

$\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$ *taking as inputs* crs*, a statement* x *and a witness* w*, outputs a proof/argument* π*.*

$\mathsf{Verify}(\mathsf{crs}, x, \pi) \to \{0,1\}$ *is deterministic, taking as inputs* crs*, a statement* x *and a proof* π*, outputs either* 1 *or* 0*.*

$\mathsf{Sim}(\mathsf{crs}, \tau_{\mathsf{sim}}, x) \to \{\pi^\star, \bot\}$ *taking as inputs* crs*, a simulation trapdoor* τ_{sim} *and statement* x*, outputs a simulated argument* $\pi^\star$ *or symbol* $\bot$ *indicating failure.*

We require the following:

Perfect completeness. For any $(x, w) \in \mathcal{R}_{\mathrm{zk}}$, it holds that

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(\lambda, \mathsf{info}_{\mathcal{L}}), \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) \end{array}\right] = 1,$$

Soundness. For every PPT adversary $\mathcal{A}$ and for all $x \notin \mathcal{L}_{\mathrm{snd}}$, it holds that

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(\lambda, \mathsf{info}_{\mathcal{L}}), \\ \pi \leftarrow \mathcal{A}(\mathsf{crs}, x) \end{array}\right] = \mathsf{negl}(\lambda).$$

Statistical zero-knowledge. For any $(\mathsf{crs}, \tau_{\mathsf{sim}}) \leftarrow \mathsf{Setup}(\lambda, \mathsf{info}_{\mathcal{L}})$, and for all $(x, w) \in \mathcal{R}_{\mathrm{zk}}$, the distributions

$$\{\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)\},$$

and

$$\{\pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \tau_{\mathsf{sim}}, x)\},$$

are statistically indistinguishable.

One-time simulation soundness [66] requires that a bounded adversary cannot produce a valid proof for a false statement, even after having seen a simulated proof of a statement of its choice.

Simulation Soundness. [66] For every PPT adversary $\mathcal{A}$ and all $x^\star \notin \mathcal{L}_{\mathrm{snd}}$,

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{crs}, x^\star, \pi^\star) = 1 \\ \quad \wedge \quad (x^\star, \pi^\star) \neq (x, \pi) \end{array}\middle|\begin{array}{r} (\mathsf{crs}, \tau_{\mathsf{sim}}) \leftarrow \mathsf{Setup}(\lambda, \mathcal{L}), \\ (x, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{crs}, x^\star), \\ \pi \leftarrow \mathsf{Sim}(\mathsf{crs}, \tau_{\mathsf{sim}}, x), \\ \pi^\star \leftarrow \mathcal{A}(\mathsf{crs}, \mathsf{st}, \pi) \end{array}\right] = \mathsf{negl}(\lambda).$$

B Simulation Sound NIZK Argument

We recall the simulation-sound NIZK argument system of [45]. The construction relies on the notions of admissible hash functions and $\mathcal{R}$-lossy encryption with efficient opening, and generically builds a simulation-sound argument system from any trapdoor Σ-protocol.

B.1 Admissible Hash Functions

Admissible hash functions [7] functions as a combinatorial tool for partitioning-based security proofs.

Definition B.1 ([7]). *Let $\ell_0(\lambda), L(\lambda) \in \mathbb{N}$ be functions of a security parameter $\lambda \in \mathbb{N}$. Let $\mathsf{AHF} : \{0,1\}^{\ell_0} \to \{0,1\}^L$ be an efficiently computable function. For every $K \in \{0,1,\perp\}^L$, let the partitioning function $F_{\mathsf{ADH}}(K,\cdot) : \{0,1\}^{\ell_0} \to \{0,1\}$ such that*

$$F_{\mathsf{ADH}}(K,X) := \begin{cases} 0 & \text{if } \forall i \in [L] \quad (\mathrm{AHF}(X)_i = K_i) \vee (K_i = \perp) \\ 1 & \text{otherwise} \end{cases}$$

*We say that AHF is an **admissible hash function** if there exists an efficient algorithm $\mathsf{AdmSmp}(1^\lambda, Q, \delta)$ that takes as inputs $Q \in \mathrm{poly}(\lambda)$ and a non-negligible $\delta(\lambda) \in (0,1]$ and outputs a key $K \in \{0,1,\perp\}^L$ such that, for all $X^{(1)}, \ldots, X^{(Q)}, X^\star \in \{0,1\}^{\ell_0}$ such that for $X^\star \notin \{X^{(1)}, \ldots, X^{(Q)}\}$, we have*

$$\Pr_K \left[F_{\mathsf{ADH}}(K, X^{(1)}) = \cdots = F_{\mathsf{ADH}}(K, X^{(Q)}) = 1 \wedge F_{\mathsf{ADH}}(K, X^\star) = 0 \right] \geq \delta(Q(\lambda)).$$

Theorem B.1 ([35, Theorem 1]). *Let $(C_{\ell_0})_{\ell_0 \in \mathbb{N}}$ be a family of codes $C_{\ell_0} : \{0,1\}^{\ell_0} \to \{0,1\}^L$ with minimal distance $c \cdot L$ for some constant $c \in (0, 1/2)$. Then, $(C_{\ell_0})_{\ell_0 \in \mathbb{N}}$ is a family of admissible hash functions. Moreover, algorithm $\mathsf{AdmSmp}(1^\lambda, Q, \delta)$ outputs a key $K \in \{0,1,\perp\}^L$ for which $\eta = O(\log \lambda)$ components are not $\perp$ and $\delta(Q(\lambda))$ is a non-negligible function of λ.*

B.2 $\mathcal{R}$-Lossy Encryption with Efficient Opening

Definition B.2 ([45]). *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy PKE scheme with efficient opening is a 7-tuple of PPT algorithms $(\mathsf{Param}, \mathsf{KeyGen}, \mathsf{LKeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \mathsf{LOpen})$ such that:*

- ***Parameter generation:** On input a security parameter λ, $\mathsf{Param}(1^\lambda)$ outputs public parameters Γ.*
- ***Key generation:** For an initialization value $K \in \mathcal{K}_\lambda$ and public parameters Γ, algorithm $\mathsf{KeyGen}(\Gamma, K)$ outputs an injective public key $pk_{\mathsf{inj}} \in \mathcal{PK}$, a decryption key $sk_{\mathsf{inj}} \in \mathcal{SK}$ and a trapdoor key $tk \in \mathcal{TK}$. The public key specifies a ciphertext space CtSp and a randomness space $\mathsf{R}^{\mathsf{LPKE}}$.*
- ***Lossy key generation:** Given an initialization value $K \in \mathcal{K}_\lambda$ and public parameters Γ, the lossy key generation algorithm $\mathsf{LKeygen}(\Gamma, K)$ outputs a lossy public key $pk_{\mathsf{lossy}} \in \mathcal{PK}$, a lossy secret key $sk_{\mathsf{lossy}} \in \mathcal{SK}$ and a trapdoor key $tk \in \mathcal{TK}$.*
- ***Decryption under injective tags:** For any initialization value $K \in \mathcal{K}$, any tag $t \in \mathcal{T}$ such that $(K,t) \in \mathcal{R}$, and any message $\mathsf{Msg} \in \mathsf{MsgSp}$, we have*

$$\Pr[\exists r \in \mathcal{R}^{\mathsf{LPKE}} : \mathsf{Dec}(sk, t, \mathsf{Encrypt}(pk, t, \mathsf{Msg}; r)) \neq \mathsf{Msg}] < \nu(\lambda),$$

for some negligible function $\nu(\lambda)$, where $(pk, sk, tk) \leftarrow \mathsf{KeyGen}(\Gamma, K)$ and the probability is taken over the randomness of KeyGen.

– ***Indistinguishability:*** *Algorithms* $\mathsf{LKeyGen}$ *and* KeyGen *satisfy the following:*
  (i) For any $K \in \mathcal{K}_\lambda$*, the distributions* $\mathsf{D}_{\mathsf{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{KeyGen}(\Gamma, K)\}$ *and* $\mathsf{D}_{\mathsf{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)\}$ *are computationally indistinguishable.*
  (ii) For any distinct initialization values $K, K' \in \mathcal{K}_\lambda$*, the two distributions* $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)\}$ *and* $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K')\}$ *are statistically indistinguishable. We require them to be* $2^{-\Omega(\lambda)}$*-close in terms of statistical distance.*
– ***Lossiness:*** *For any initialization value* $K \in \mathcal{K}_\lambda$ *and tag* $t \in \mathcal{T}_\lambda$ *such that* $(K, t) \notin \mathcal{R}$*, any* $(pk, sk, tk) \leftarrow \mathsf{KeyGen}(\Gamma, K)$*, and any* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, the following distributions are statistically close:*

$$\{C \mid C \leftarrow \mathsf{Enc}(pk, t, \mathsf{Msg}_0)\} \approx_s \{C \mid C \leftarrow \mathsf{Enc}(pk, t, \mathsf{Msg}_1)\}.$$

  For any $(pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)$*, the above holds for any tag* t *(and not only those for which* $(K, t) \notin \mathcal{R}$*).*
– ***Efficient opening under lossy tags:*** *Let* D_R *denote the distribution, defined over the randomness space* $\mathsf{R}^{\mathsf{LPKE}}$*, from which the random coins used by* Enc *are sampled. For any message* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext* C*, let* $\mathsf{D}_{PK,\mathsf{Msg},C,t}$ *denote the probability distribution on* $\mathsf{R}^{\mathsf{LPKE}}$ *with support*

$$\mathsf{S}_{PK,\mathsf{Msg},C,t} = \{\overline{r} \in \mathsf{R}^{\mathsf{LPKE}} \mid \mathsf{Enc}(pk, t, \mathsf{Msg}, \overline{r}) = C\},$$

  and such that, for each $\overline{r} \in \mathsf{S}_{PK,\mathsf{Msg},C,t}$*, we have*

$$\mathsf{D}_{PK,\mathsf{Msg},C,t}(\overline{r}) = \Pr_{r' \leftarrow \mathsf{D}_\mathsf{R}}[r' = \overline{r} \mid \mathsf{Enc}(pk, t, \mathsf{Msg}, r') = C].$$

  There exists a PPT algorithm Open *such that, for any* $K \in \mathcal{K}_\lambda$*, any keys* $(pk, sk, tk) \leftarrow \mathsf{KeyGen}(\Gamma, K)$ *and* $(pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)$*, any random coins* $r \leftarrow \mathsf{D}_\mathsf{R}$*, any tag* $t \in \mathcal{T}_\lambda$ *such that* $(K, t) \notin \mathcal{R}$*, and any messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, takes as inputs* $pk, C = \mathsf{Enc}(pk, t, \mathsf{Msg}_0, r)$*,* t *and* tk*, and outputs a sample* $\overline{r}$ *from a distribution statistically close to* $\mathsf{D}_{PK,\mathsf{Msg}_1,C,t}$*.*
– ***Efficient opening under lossy keys:*** *There exists a PPT sampling algorithm* LOpen *such that, for any* $K \in \mathcal{K}_\lambda$*, any keys* $(pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)$*, any random coins* $r \leftarrow \mathsf{D}_\mathsf{R}$*, any tag* $t \in \mathcal{T}_\lambda$*, and any distinct messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, takes as inputs* $C = \mathsf{Enc}(pk, t, \mathsf{Msg}_0, r)$*,* t *and* sk*. It outputs a sample* $\overline{r}$ *from a distribution statistically close to* $\mathsf{D}_{PK,\mathsf{Msg}_1,C,t}$*.*

Like [12], [45] considers $\mathcal{R}$-lossy PKE schemes for the bit-matching relation, which is defined from the admissible hash function as in Appendix B.1

Definition B.3. *Let* $\mathcal{K} = \{0, 1, \perp\}^L$ *and* $\mathcal{T} = \{0, 1\}_0^\ell$*, for some* $\ell_0, L \in poly(\lambda)$ *such that* $\ell_0 < L$*. Let* F_{ADH} *be the partition function defined by an admissible hash function* $\mathsf{AHF} : \{0, 1\}_0^\ell \to \{0, 1\}^L$*. The bit-matching relation* $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0, 1\}$ *for* AHF *is defined as* $\mathcal{R}_{\mathsf{BM}}(K, t) = 1$ *if and only if* $K = K_1 \dots K_L$ *and* $t = t_1 \dots t_{\ell_0}$ *satisfy* $F_{\mathsf{ADH}}(K, t) = 0$*, i.e.,* $\bigwedge_{i=1}^{L}(K_i = \perp) \vee (K_i = \mathsf{AHF}(t)[i])$*.*

We refer to [45] for a lattice-based construction of $\mathcal{R}_{\mathsf{BM}}$-lossy PKE. The proposed encryption scheme is a variant of the primal-Regev encryption scheme [65] suggested in [31].

B.3 Construction of One-Time Simulation-Sound NIZK

The compiler of [45] turns any trapdoor Σ-protocol into an *unbounded* simulation-sound non-interactive argument. For CCA-secure group signatures, as one-time simulation-sound is sufficient, we use the syntax of [45] but remove the labels associating to the statements.

The construction relies on the following building blocks:

- A trapdoor Σ-protocol $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{TrapGen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ with challenge space $\mathcal{C}$, for a language $\mathcal{L} = (\mathcal{L}_{\mathrm{zk}}, \mathcal{L}_{\mathrm{snd}})$. We assume that the $\mathsf{BadChallenge}$ algorithm underlying Π should be computable within time $T \in \mathsf{poly}(\lambda)$;
- A somewhere CI hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ with output length $\kappa \in \mathsf{poly}(\lambda)$ for the class $\mathcal{R}_{\mathrm{CI}}$ of relations that are efficiently searchable within time T. In particular, we assume that the $\mathsf{BadChallenge}$ function underlying Π is computable in time T;
- A strongly unforgeable one-time signature scheme $\Pi^{\mathsf{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell_0 \in \mathrm{poly}(\lambda)$.
- An admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_0} \to \{0,1\}^L$, where $L \in \mathsf{poly}(\lambda)$ such that $\ell_0 < L$. The function induces the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^{\ell_0} \to \{0,1\}$ of Definition B.3;
- An $\mathcal{R}$-lossy public-key encryption scheme

$$\mathcal{R}\text{−}\mathsf{LPKE} = (\mathsf{Param}, \mathsf{KeyGen}, \mathsf{LKeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \mathsf{LOpen})$$

  for the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^{\ell_0} \to \{0,1\}$ with public (resp. secret) key space $\mathcal{PK}$ (resp. $\mathcal{SK}$). We assume that the decryption algorithm Dec is computable within time T. We denote the message (resp. ciphertext) space by MsgSp (resp. CtSp) and the randomness space by $\mathsf{R}^{\mathsf{LPKE}}$. Let also $\mathsf{D}_{\mathsf{R}^{\mathsf{LPKE}}}$ denote the distribution from which the random coins of Enc are sampled.

We also assume that these ingredients are compatible in the sense that P outputs a first prover message that fits in the message space MsgSp of $\mathcal{R}$-LPKE. Our construction $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim})$ goes as follows.

- $\mathsf{Setup}(1^\lambda, \mathsf{info}_{\mathcal{L}})$: taking as inputs a security parameter λ and language-specific information $\mathsf{info}_{\mathcal{L}}$, does the following:
  1. Run $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ to obtain $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$.
  2. Generate a common reference string $\mathsf{crs}_\Sigma = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ for the trapdoor Σ-protocol Π, by computing its language-dependent part as $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathsf{info}_{\mathcal{L}})$.
  3. Choose a strongly unforgeable one-time signature scheme $\Pi^{\mathsf{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell_0 \in \mathrm{poly}(\lambda)$.
  4. Generate public parameters $\Gamma \leftarrow \mathsf{Param}(1^\lambda)$ for the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme where the relation $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is defined by an admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_0} \to \{0,1\}^L$. Choose a random initialization value $K \leftarrow \mathcal{K}$ and generate lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeyGen}(\Gamma, K)$.
  5. Generate a key $k \leftarrow \mathrm{Gen}(1^\lambda)$ for the somewhere CI hash function.
  6. Choose a number of parallel repetitions $\kappa = \mathsf{poly}(\lambda)$.

Output the common reference string as

$$\mathsf{crs} = (\mathsf{crs}_\Sigma, \mathsf{AHF}, \Pi^{\mathsf{ots}}, \Gamma, pk, k, \kappa),$$

and simulation trapdoor $\tau_{\mathsf{sim}} = sk$.

- $\mathsf{Prove}(\mathsf{crs}, x, w)$: To prove a statement $x \in \mathcal{L}_{\mathrm{zk}}$ using a witness $w \in \mathcal{R}_{\mathrm{zk}}(x)$, generate a one-time key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \Pi^{\mathsf{ots}}.\mathcal{G}(1^\lambda)$ and do the following.
  1. Compute $(\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa), \mathsf{st})$ via κ invocations of the prover P of Π;
  2. For each $i \in [\kappa]$, compute $\mathbf{a}'_i \leftarrow \mathsf{Enc}(pk, \mathsf{VK}, \mathbf{a}_i; r_i)$ using randomness $r_i \leftarrow \mathsf{D}_{\mathsf{R}^{\mathsf{LPKE}}}$. Define $\mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa)$ and $\mathbf{r} = (r_1, \ldots, r_\kappa)$.
  3. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}', \mathsf{VK})) \in \mathcal{C}^\kappa$;
  4. Compute $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\kappa) \leftarrow \mathsf{P}(\mathsf{crs}_\Sigma, x, w, \mathbf{a}, \mathsf{Chall}, \mathsf{st})$ via κ invocation of prover of Π. Let $\mathbf{z}' = (\mathbf{z}, \mathbf{a}, \mathbf{r})$;
  5. Generate a one-time signature $\mathsf{sig} \leftarrow \Pi^{\mathsf{ots}}.\mathcal{S}(\mathsf{SK}, x, \mathbf{a}', \mathbf{z}')$ and output the proof $\pi = (\mathsf{VK}, \mathbf{a}', \mathbf{z}', \mathsf{sig})$.
- $\mathsf{Verify}(\mathsf{crs}, x, \pi)$: Given a statement x and a candidate proof $\pi = (\mathsf{VK}, \mathbf{a}', \mathbf{z}', \mathsf{sig})$, return 0 if $\Pi^{\mathsf{ots}}.\mathcal{V}(\mathsf{VK}, (x, \mathbf{a}', \mathbf{z}'), \mathsf{sig}) = 0$. Otherwise, proceed as follows:
  1. Parse $\mathbf{z}'$ as $\mathbf{z}' = (\mathbf{z}, \mathbf{a}, \mathbf{r}) = ((\mathbf{z}_1, \ldots, \mathbf{z}_\kappa), (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa), (r_1, \ldots, r_\kappa))$, parse $\mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa)$ and return 0 if it does not parse properly. Return 0 if there exists $i \in [\kappa]$ such that $\mathbf{a}'_i \neq \mathsf{Enc}(pk, \mathsf{VK}, \mathbf{a}_i; r_i)$ or $r_i \notin \mathsf{R}^{\mathsf{LPKE}}$;
  2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}', \mathsf{VK})) \in \mathcal{C}^\kappa$. Then invoke the verifier V of Π to check if $\mathsf{V}(\mathsf{crs}_\Sigma, x, (\mathbf{a}_i, \mathsf{Chall}[i], \mathbf{z}_i)) = 1$ for each $i \in [\kappa]$. If the check succeeds, return 1; else return 0.
- $\mathsf{Sim}(\mathsf{crs}, \tau_{\mathsf{sim}}, x)$: given a common reference string crs, a statement x and a simulation trapdoor $\tau_{\mathsf{sim}} = sk$ as input, the simulator does the following:
  1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Let $\mathbf{0}^{|\mathbf{a}|}$ the all-zeroes string of length $|\mathbf{a}|$. Sample random coins $\mathbf{r}_0 \leftarrow \mathsf{D}_{\mathsf{R}}^{\mathsf{LPKE}}$ from the distribution $\mathsf{D}_{\mathsf{R}}^{\mathsf{LPKE}}$ and compute $\mathbf{a}' \leftarrow \mathsf{Enc}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}|}; \mathbf{r}_0)$.
  2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK}))$;
  3. Run the special ZK simulator $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}_\Sigma, x, \mathsf{Chall})$ of Π to obtain a simulated transcript $(\mathbf{a}, \mathbf{z})$ of Π for the challenge Chall;
  4. Using the lossy secret key sk of $\mathcal{R}$-LPKE, compute random coins $\mathbf{r} \leftarrow \mathsf{Open}(sk, \mathsf{VK}, \mathbf{a}', \mathbf{a})$ which explains $\mathbf{a}'$ as an encryption of $(x, \mathbf{a})$ under the tag VK. Then set $\mathbf{z}' = (\mathbf{z}, \mathbf{a}, \mathbf{r})$;
  5. Compute $\mathsf{sig} \leftarrow \mathcal{S}(\mathsf{sk}, x, \mathbf{a}', \mathbf{z}')$ and output $\pi = (\mathsf{VK}, (\mathbf{a}', \mathbf{z}'), \mathsf{sig})$.

Theorem B.2 ([45]). *The above NIZK system is complete, multi-theorem zero-knowledge and one-time simulation sound, assume that: (i) Π^{ots} is strongly unforgeable, (ii) $\mathcal{R}$-LPKE is a $\mathcal{R}_{\mathsf{BM}}$-lossy encryption scheme, (iii) the LWE assumption holds, (iv) the hash family $\mathcal{H}$ is correlation-intractable for all relations that are searchable within time T and (v) the underlying trapdoor protocol Π is special zero-knowledge.*

C Deferred Security Proof

C.1 Anonymity

Theorem C.1. *Assume that: (i) $\mathsf{LWE}_{n,q,\chi}$ is hard against PPT adversaries; (ii) the NIZK system is one-time simulation-sound and statistically zero-knowledge;*

and (iii) $\mathsf{SIS}_{q,n,\ell_{\mathbf{m}}+2m,B_{\mathsf{anon}}}$ *is hard, where* $B_{\mathsf{anon}} = 2\sigma_{\mathsf{sign}}\sqrt{2m} + \sqrt{\ell_{\mathbf{m}}}$*. Then the above group signature scheme is CCA-fully anonymous.*

Proof. We prove the result using a sequence of games. In the first game, the challenger runs experiment $\mathbf{Exp}^{\mathsf{anon-b}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ as in **Fig.** 3, where $b \in \{0,1\}$ is fixed. In the last game, the view of the adversary is independent of b. For $b \in \{0,1\}$, we write **Game** i_b to denote **Game** i where the challenger's bit is b, and let $p_{i,b}$ be the probability that the adversary outputs $b' = 1$ in **Game** i_b. We recall that $\mathcal{A}$ may query the opening oracle on any pair $(\mathbf{m}, \Sigma) \neq (\mathbf{m}^\star, \Sigma^\star)$, where $(\mathbf{m}^\star, \Sigma^\star)$ is the challenge.

For a message-signature pair $(\mathbf{m}, \Sigma)$ with $\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)$, we call $\mathsf{st} = (\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$, computed as in Verify, the statement of $(\mathbf{m}, \Sigma)$. We write $\mathsf{st}^\star$ for the statement of $(\mathbf{m}^\star, \Sigma^\star)$.

Game 0_b**:** This is the real experiment $\mathbf{Exp}^{\mathsf{anon-b}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ defining CCA-full anonymity in **Fig.** 3. The challenger runs KeyGen and gives $(\mathsf{gpk}, \{\mathsf{gsk}[\mathsf{id}]\}_{\mathsf{id}\in[N]})$ to $\mathcal{A}$. Opening queries are answered by running $\mathsf{Open} = \mathsf{Open}^{(1)}$. In the challenge phase, $\mathcal{A}$ chooses $\mathsf{id}_0^\star, \mathsf{id}_1^\star \in [N]$ and a message $\mathbf{m}^\star \in \{0,1\}^{\ell_{\mathbf{m}}}$, and obtains $\Sigma^\star = (\mathbf{s}_1^\star, \mathbf{s}_2^\star, \mathbf{c}_1^\star, \mathbf{c}_2^\star, \pi^\star) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[\mathsf{id}_b^\star], \mathbf{m}^\star)$. Let $\mathsf{id}^\star = \mathsf{id}_b^\star$, let $\tau^\star$ be the tag used by Sign, let $\mathbf{x}^\star = (\mathbf{s}^\star, \mathbf{s}_1^\star, \mathbf{s}_2^\star)$ be the JRS signature sampled in Sign, and let $\mathbf{p}^\star = (\mathbf{s}^\star, \mathsf{bin}(\mathsf{id}^\star), \mathsf{bin}(\tau^\star))$ be the plaintext of $\mathbf{c}_1^\star$ and $\mathbf{c}_2^\star$. Note that by definition, $p_{0,b} = \Pr[\mathbf{Exp}^{\mathsf{anon-b}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = 1]$ and $\mathsf{Adv}^{\mathsf{anon}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = |p_{0,0} - p_{0,1}|$.

Game 1_b**:** We change the way $\mathbf{A}$, $\mathbf{B}$ and the users' secret keys are generated, and the way the vector $\mathbf{x}^\star$ is sampled. The challenger runs $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q, \mathbf{I}_n)$ (Lemma 2.3) and samples $\mathbf{B} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$. For each $\mathsf{id} \in [N]$, it computes each column $\mathbf{R}_{\mathsf{id}}[h]$ of $\mathsf{gsk}[\mathsf{id}]$ independently as follows: it samples $\mathbf{v}_2 \leftarrow \mathcal{D}_\sigma^m$, then samples $\mathbf{v}_1 \leftarrow \mathsf{SampleD}(\mathbf{T}_{\mathbf{A}}, \mathbf{A}, \mathbf{C}[h] - (\mathsf{id}\cdot\mathbf{G}_{n,m} - \mathbf{B})\cdot\mathbf{v}_2, \sigma)$ (Lemma 2.4) and sets $\mathbf{R}_{\mathsf{id}}[h] = (\mathbf{v}_1, \mathbf{v}_2)$. To compute $\Sigma^\star$, it samples $(\mathbf{s}_1^\star, \mathbf{s}_2^\star) \leftarrow \mathcal{D}_{\sigma_{\mathsf{sign}}}^{2m}$, then

$$\mathbf{s}^\star \leftarrow \mathsf{SampleD}\left(\mathbf{T}_{\mathbf{A}}, \mathbf{A}, \mathbf{u} + \mathbf{D}\cdot\mathbf{m}^\star - (\mathsf{id}^\star\cdot\mathbf{G}_{n,m} - \mathbf{B} \mid \tau^\star\cdot\mathbf{G}_{n,m} - \mathbf{C})\cdot(\mathbf{s}_1^\star, \mathbf{s}_2^\star), \sigma_{\mathsf{sign}}\right),$$

and performs the remaining steps of Sign faithfully.

These calls to $\mathsf{SampleD}$ are possible as $\|\mathbf{T}_{\mathbf{A}}\|_2 \leq m$ by Lemma 2.3, and $\sigma, \sigma_{\mathsf{sign}} \geq \mathcal{O}(1)\cdot\sqrt{1+m^2}$. The distributions of $(\mathbf{A}, \mathbf{B})$ in **Game 0** and in **Game 1** are both statistically close to uniform, by Lemma 2.2 and Lemma 2.3. By Lemma 2.4 and Lemma 2.5, the columns of each $\mathsf{gsk}[\mathsf{id}]$ and the vector $\mathbf{x}^\star$ have statistically close distributions in both games. Therefore $|p_{1,b} - p_{0,b}| \leq \mathsf{negl}(\lambda)$. In **Game** 1_b, $(\mathbf{s}_1^\star, \mathbf{s}_2^\star)$ is independent of $\mathsf{id}^\star$, $\tau^\star$ and b. Furthermore, the challenger does not use any trapdoor of $\mathbf{B}$ or $\mathbf{C}$, and the matrix $(\mathbf{D} \mid \mathbf{B} \mid \mathbf{C}) \in \mathbb{Z}_q^{n\times(\ell_{\mathbf{m}}+2m)}$ is uniform and independent of $\mathbf{A}$.

Game 2_b**:** This game is identical to **Game** 1_b, except that the challenger generates crs with the simulator of the NIZK system, retaining the decryption keys $(\mathbf{S}_1, \mathbf{S}_2)$ as in KeyGen, and generates the proof $\pi^\star$ in the challenge signature $\Sigma^\star$

by running the NIZK simulator on $\mathsf{st}^\star$. The statistical zero-knowledge property of the NIZK system implies that $|p_{2,b} - p_{1,b}| \leq \mathsf{negl}(\lambda)$.

Game 3_b**:** This game is identical to **Game** 2_b, except that the challenger answers opening queries by running $\mathsf{Open}^{(2)}$, i.e., the algorithm Open with the decryption key $\mathbf{S}_2$ and the ciphertext $\mathbf{c}_2$.

We show that the two games are computationally indistinguishable. Assume otherwise, then with non-negligible probability $\mathcal{A}$ queries a valid pair $(\mathbf{m}, \Sigma)$ with $\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)$ to the opening oracle, such that the outputs of $\mathsf{Open}^{(1)}$ and $\mathsf{Open}^{(2)}$ differ. Let st be the statement of $(\mathbf{m}, \Sigma)$. By Proposition 4.2, $\mathsf{st} \notin \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}}$. We distinguish two cases:

- If $\mathcal{A}$ makes this query before seeing $(\mathbf{m}^\star, \Sigma^\star)$, it directly violates the soundness of the NIZK system.
- If $\mathcal{A}$ makes this query after seeing $(\mathbf{m}^\star, \Sigma^\star)$, then $(\mathbf{m}, \Sigma) \neq (\mathbf{m}^\star, \Sigma^\star)$. If $(\mathsf{st}, \pi) \neq (\mathsf{st}^\star, \pi^\star)$, then $\mathcal{A}$ creates a valid proof π for the false statement st after seeing the simulated proof $\pi^\star$ of $\mathsf{st}^\star$, which violates the one-time simulation-soundness of the NIZK system.
  Otherwise, $(\mathsf{st}, \pi) = (\mathsf{st}^\star, \pi^\star)$. Then $\mathbf{t} = \mathbf{t}^\star$, and $(\mathbf{m}, \mathbf{s}_1, \mathbf{s}_2) \neq (\mathbf{m}^\star, \mathbf{s}_1^\star, \mathbf{s}_2^\star)$ since the pairs $(\mathbf{m}, \Sigma)$ and $(\mathbf{m}^\star, \Sigma^\star)$ differ. It follows that

  $$(\mathbf{D} \mid \mathbf{B} \mid \mathbf{C}) \cdot \mathbf{z} = \mathbf{0}^n \bmod q, \quad \mathbf{z} = (\mathbf{m} - \mathbf{m}^\star, \mathbf{s}_1 - \mathbf{s}_1^\star, \mathbf{s}_2 - \mathbf{s}_2^\star) \in \mathbb{Z}^{\ell_{\mathbf{m}}+2m} \backslash \{\mathbf{0}\}.$$

  As $(\mathbf{m}, \Sigma)$ is valid and $\|(\mathbf{s}_1^\star, \mathbf{s}_2^\star)\|_2 \leq \sigma_{\mathsf{sign}}\sqrt{2m}$ except with negligible probability (Lemma 2.1), we have that $\|\mathbf{z}\|_2 \leq 2\sigma_{\mathsf{sign}}\sqrt{2m} + \sqrt{\ell_{\mathbf{m}}} = B_{\mathsf{anon}}$. Since $(\mathbf{D} \mid \mathbf{B} \mid \mathbf{C})$ is uniform and the challenger only uses the trapdoor $\mathbf{T}_{\mathbf{A}}$ of the independent matrix $\mathbf{A}$ to generate gsk and $\mathbf{x}^\star$, such an adversary yields a solution to $\mathsf{SIS}_{q,n,\ell_{\mathbf{m}}+2m,B_{\mathsf{anon}}}$.

Therefore we have that $|p_{3,b} - p_{2,b}| \leq \mathsf{negl}(\lambda)$.

Game 4_b**:** This game is identical to **Game** 3_b, except that the challenger samples $\mathbf{V}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times(m+k+\ell)})$ instead of computing $\mathbf{V}_1 = \mathbf{U}_1\mathbf{S}_1 \bmod q$. Note that the decryption key $\mathbf{S}_1$ is not used since **Game** 3_b. By Lemma 2.2, $|p_{4,b} - p_{3,b}| \leq 2^{-\Omega(\lambda)}$.

Game 5_b**:** This game is identical to **Game** 4_b, except that the challenger computes $\mathbf{c}_1^\star = \mathbf{w}_1 + (\mathbf{0}^m, \lfloor q/2K \rfloor \cdot \mathbf{p}^\star) \bmod q$ for $\mathbf{w}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{2m+k+\ell})$, instead of faithfully generating $\mathbf{c}_1^\star$ as a dual-Regev ciphertext of $\mathbf{p}^\star$. Thus $\mathbf{c}_1^\star$ is uniformly distributed and independent of $\mathbf{p}^\star$. As $(\mathbf{U}_1, \mathbf{V}_1)$ is uniform and $\pi^\star$ is simulated, we have that $|p_{5,b} - p_{4,b}| \leq \mathsf{Adv}_{\mathsf{LWE}_{n,q,\chi}}$.

Game 6_b**:** This game is identical to **Game** 5_b, except that the challenger samples $\mathbf{S}_1 \leftarrow \mathcal{U}(\{0,1\}^{m\times(m+k+\ell)})$ and computes $\mathbf{V}_1 = \mathbf{U}_1\mathbf{S}_1 \bmod q$, as in **Game** 3_b. By Lemma 2.2, $|p_{6,b} - p_{5,b}| \leq 2^{-\Omega(\lambda)}$.

Game 7_b**:** This game is identical to **Game** 6_b, except that the challenger answers opening queries by running $\mathsf{Open}^{(1)}$ again. In both games, $\mathbf{V}_i = \mathbf{U}_i\mathbf{S}_i \bmod q$ for $i \in \{1,2\}$, and the two games are identical unless $\mathcal{A}$ queries a valid pair $(\mathbf{m}, \Sigma)$ to the opening oracle such that the outputs of $\mathsf{Open}^{(1)}$ and $\mathsf{Open}^{(2)}$ differ. Similar to the transition from **Game** 2_b to **Game** 3_b, we have that $|p_{7,b} - p_{6,b}| \leq \mathsf{negl}(\lambda)$.

Game 8_b**:** This game is identical to **Game** 7_b, except that the challenger samples $\mathbf{V}_2 \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times(m+k+\ell)})$ instead of computing $\mathbf{V}_2 = \mathbf{U}_2\mathbf{S}_2 \bmod q$. Note that the decryption key $\mathbf{S}_2$ is not used since **Game** 7_b. By Lemma 2.2, $|p_{8,b} - p_{7,b}| \leq 2^{-\Omega(\lambda)}$.

Game 9_b**:** This game is identical to **Game** 8_b, except that the challenger computes $\mathbf{c}_2^\star = \mathbf{w}_2 + (\mathbf{0}^m, \lfloor q/2K \rfloor \cdot \mathbf{p}^\star) \bmod q$ for $\mathbf{w}_2 \leftarrow \mathcal{U}(\mathbb{Z}_q^{2m+k+\ell})$. As in the transition from **Game** 4_b to **Game** 5_b, with the opening queries now answered with $\mathbf{S}_1$, we have that $|p_{9,b} - p_{8,b}| \leq \mathsf{Adv}_{\mathsf{LWE}_{n,q,\chi}}$.

In **Game** 9_b, the challenge signature $\Sigma^\star = (\mathbf{s}_1^\star, \mathbf{s}_2^\star, \mathbf{c}_1^\star, \mathbf{c}_2^\star, \pi^\star)$ is distributed independently of b. Indeed, $(\mathbf{s}_1^\star, \mathbf{s}_2^\star)$ is sampled independently of b since **Game** 1_b, the ciphertexts $\mathbf{c}_1^\star$ and $\mathbf{c}_2^\star$ are uniform over $\mathbb{Z}_q^{2m+k+\ell}$, and $\pi^\star$ is a simulated proof of $\mathsf{st}^\star$, none of whose components depends on b. Moreover, gpk, gsk and the answers of the opening oracle are computed independently of b. Hence $p_{9,0} = p_{9,1}$. Since $|p_{i+1,b} - p_{i,b}| \leq \mathsf{negl}(\lambda)$ for each $b \in \{0,1\}$ and each $i \in \{0,\ldots,8\}$, we have that $\mathsf{Adv}^{\mathsf{anon}}_{\mathsf{GS},\mathcal{A}}(\lambda, N) = |p_{0,0} - p_{0,1}| \leq \mathsf{negl}(\lambda)$. □

C.2 Traceability

Throughout this section, let $l = m + k + \ell$. An adversary $\mathcal{A}$ in the experiment $\mathbf{Exp}^{\mathsf{trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, N)$ (**Fig.** 2) wins by outputting a valid pair $(\mathbf{m}^\star, \Sigma^\star)$ such that either Open returns $\perp$, or Open returns an identity $\mathsf{id}^\star \notin \mathsf{CU}$ such that $(\mathsf{id}^\star, \mathbf{m}^\star)$ was not queried to the signing oracle. In this section, we refer to the former as a *type-1 forgery*, and to the latter as a *type-2 forgery*. To show that the scheme is traceable, we treat each type of forgery in a separate theorem.

Extraction from Forgery. We first describe what the reduction can extract from a forgery. For a vector $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \in \mathbb{Z}^m \times \mathbb{Z}^k \times \mathbb{Z}^\ell$, we write

$$a_{\mathsf{id}}(\mathbf{p}) = \sum_{i\in[k]} \mathbf{p}_2[i] \cdot 2^{i-1} \in \mathbb{Z}, \quad a_\tau(\mathbf{p}) = \sum_{i\in[\ell]} \mathbf{p}_3[i] \cdot 2^{i-1} \in \mathbb{Z},$$

and simply a_{id}, a_τ when $\mathbf{p}$ is clear from the context. Let $\mathsf{Ext}(\mathbf{S}_1, \mathbf{m}, \Sigma)$ be the deterministic algorithm that runs $\mathsf{Open}(\mathsf{gpk}, \mathbf{S}_1, \mathbf{m}, \Sigma)$ and proceeds as follows. If Open returns an identity, then Ext outputs the pair $(\mathbf{p}, c)$ of the iteration at which Open returns. Otherwise, Ext outputs the pair $(\mathbf{p}, c)$ of the first iteration that passes Step (III), or $\perp$ if there is no such iteration.

Lemma C.1. *Let* $\mathbf{m} \in \{0,1\}^{\ell_\mathbf{m}}$, $\Sigma = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c}_1, \mathbf{c}_2, \pi)$ *such that* $\mathsf{Verify}(\mathsf{gpk}, \mathbf{m}, \Sigma) = 1$, *and let* $\mathsf{st} = (\mathbf{b}, \mathbf{d}, \mathbf{t}, \mathbf{c}_1, \mathbf{c}_2)$ *be the statement constructed from* $(\mathbf{m}, \Sigma)$. *If*

$\mathsf{Open}(\mathsf{gpk}, \mathbf{S}_1, \mathbf{m}, \Sigma) \neq \perp$ *or* $\mathsf{st} \in \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}}$*, then* $\mathsf{Ext}(\mathbf{S}_1, \mathbf{m}, \Sigma)$ *outputs a pair* $(\mathbf{p}, c)$ *with* $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ *and* $c \in [C]$ *such that* $\|\mathbf{p}\|_2 \leq 2\alpha\sqrt{l}$*,* $|a_{\mathsf{id}}| \leq 2\alpha\sqrt{l} \cdot N$*,* $|a_\tau| \leq 2\alpha\sqrt{l} \cdot q_{\mathsf{tag}}$*, and*

$$\mathbf{A} \cdot \mathbf{p}_1 + (a_{\mathsf{id}} \cdot \mathbf{G} - c \cdot \mathbf{B}) \cdot \mathbf{s}_1 + (a_\tau \cdot \mathbf{G} - c \cdot \mathbf{C}) \cdot \mathbf{s}_2 = c \cdot (\mathbf{u} + \mathbf{D} \cdot \mathbf{m}) \bmod q, \tag{31}$$

where $\mathbf{G} = \mathbf{G}_{n,m}$*. Let* $\mathsf{id}_{\mathsf{ext}} = c^{-1} \cdot a_{\mathsf{id}} \bmod q$*. If* $\mathsf{Open}(\mathsf{gpk}, \mathbf{S}_1, \mathbf{m}, \Sigma)$ *returns* $\perp$*, then* $\mathsf{id}_{\mathsf{ext}} \notin [N]$*. If* $\mathsf{Open}(\mathsf{gpk}, \mathbf{S}_1, \mathbf{m}, \Sigma)$ *returns* $\mathsf{id} \in [N]$*, then* $\mathsf{id}_{\mathsf{ext}} = \mathsf{id}$*.*

Proof. If Open returns an identity, then the corresponding iteration passes Step (III). If $\mathsf{st} \in \mathcal{L}^{\mathsf{cca}}_{\mathrm{snd}}$, then Claim 2 in the proof of Proposition 4.2, applied to $\mathsf{Open}^{(1)} = \mathsf{Open}$, shows that some iteration passes Step (III). Hence, Ext outputs a pair $(\mathbf{p}, c)$ with $c \in [C]$, such that $\mathbf{p}$ passes Step (III) at iteration c. Then $\|\mathbf{p}\|_2 \leq 2\alpha\sqrt{l}$, and $|a_{\mathsf{id}}| \leq \|\mathbf{p}_2\|_\infty \cdot \sum_{i \in [k]} 2^{i-1} \leq 2\alpha\sqrt{l} \cdot N$. Similarly, $|a_\tau| \leq 2\alpha\sqrt{l} \cdot q_{\mathsf{tag}}$. Expanding the equation of Step (III) with $\mathbf{b} = \mathbf{G}\mathbf{s}_1 \bmod q$, $\mathbf{d} = \mathbf{G}\mathbf{s}_2 \bmod q$ and $\mathbf{t} = \mathbf{u} + \mathbf{D}\mathbf{m} + \mathbf{B}\mathbf{s}_1 + \mathbf{C}\mathbf{s}_2 \bmod q$ gives (31).

If Open returns $\mathsf{id} \in [N]$, then a_{id} passes Step (IV) at the iteration c, meaning that $a_{\mathsf{id}} = c \cdot \mathsf{id} \bmod q$ and thus $\mathsf{id}_{\mathsf{ext}} = \mathsf{id}$ since $N < q/2$. Otherwise, if Open returns $\perp$, then the iteration c fails Step (IV), i.e., $a_{\mathsf{id}} \neq c \cdot \mathsf{id} \bmod q$ for all $\mathsf{id} \in [N]$, so $\mathsf{id}_{\mathsf{ext}} \notin [N]$. □

When Ext is run on the forgery $(\mathbf{m}^\star, \Sigma^\star)$, we mark every extracted object with $\star$, i.e., we write $(\mathbf{p}^\star, c^\star) = \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$, $\mathbf{p}^\star = (\mathbf{p}_1^\star, \mathbf{p}_2^\star, \mathbf{p}_3^\star)$, $a_{\mathsf{id}}^\star = a_{\mathsf{id}}(\mathbf{p}^\star)$, $a_\tau^\star = a_\tau(\mathbf{p}^\star)$, $\mathsf{id}_{\mathsf{ext}}^\star = (c^\star)^{-1} \cdot a_{\mathsf{id}}^\star \bmod q$ and $\tau_{\mathsf{ext}}^\star = (c^\star)^{-1} \cdot a_\tau^\star \bmod q$, which is the "tag" extracted from the forgery. We stress that neither $\mathsf{id}_{\mathsf{ext}}^\star$ nor $\tau_{\mathsf{ext}}^\star$ needs to be well-formed. However, by Lemma C.1, they always belong to the sets

$$S_{\mathsf{id}} = \left\{ a \cdot c^{-1} \bmod q : a \in \mathbb{Z},\ |a| \leq 2\alpha\sqrt{l} \cdot N,\ c \in [C] \right\},$$
$$S_\tau = \left\{ a \cdot c^{-1} \bmod q : a \in \mathbb{Z},\ |a| \leq 2\alpha\sqrt{l} \cdot q_{\mathsf{tag}},\ c \in [C] \right\},$$

respectively, where $|S_{\mathsf{id}}| \leq C \cdot (4\alpha\sqrt{l} \cdot N + 1)$ and $|S_\tau| \leq C \cdot (4\alpha\sqrt{l} \cdot q_{\mathsf{tag}} + 1)$ are both polynomial in λ. Hence a reduction can guess $\mathsf{id}_{\mathsf{ext}}^\star$ and $\tau_{\mathsf{ext}}^\star$ with inverse-polynomial probability.

The Setup. The reductions receive a matrix $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$ from an SIS instance and choose two guesses $\overline{\mathsf{id}}, \overline{\tau} \in \mathbb{Z}_q$. They sample $\mathbf{T}_1, \mathbf{T}_2 \leftarrow \mathcal{U}(\{0,1\}^{m \times m})$ and $\mathbf{T}_3 \leftarrow \mathcal{U}(\{-1,0,1\}^{m \times \ell_{\mathbf{m}}})$, and set

$$\mathbf{B} = \mathbf{A}\mathbf{T}_1 + \overline{\mathsf{id}} \cdot \mathbf{G} \bmod q, \quad \mathbf{C} = \mathbf{A}\mathbf{T}_2 + \overline{\tau} \cdot \mathbf{G} \bmod q, \quad \mathbf{D} = \mathbf{A}\mathbf{T}_3 \bmod q. \tag{32}$$

By Lemma 2.2, $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is statistically close to uniform for any fixed $(\overline{\mathsf{id}}, \overline{\tau})$. Moreover, for every identity id, we have $\mathbf{A}_{\mathsf{id}} = (\mathbf{A} \mid (\mathsf{id} - \overline{\mathsf{id}}) \cdot \mathbf{G} - \mathbf{A}\mathbf{T}_1)$. Thus, if $\mathsf{id} \neq \overline{\mathsf{id}} \bmod q$, then $\mathbf{T}_1$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}_{\mathsf{id}}$ with tag $(\mathsf{id} - \overline{\mathsf{id}}) \cdot \mathbf{I}_n$. The reduction can compute $\mathsf{gsk}[\mathsf{id}] = \mathbf{R}_{\mathsf{id}}$ column-wise by running $\mathsf{SampleD}$ (Lemma 2.4)

with this trapdoor, exactly as in KeyGen, since $\|\mathbf{T}_1\|_2 \leq m$. For such users, the reduction answers corruption and signing queries honestly with $\mathbf{R}_{\mathsf{id}}$. Recall from KeyGen that $\mathbf{R}_{\mathsf{id}}$ is a $\mathbf{G}$-trapdoor of $(\mathbf{A}_{\mathsf{id}} \mid \tau \cdot \mathbf{G} - \mathbf{C})$ for every tag $\tau \in [q_{\mathsf{tag}}]$, regardless of $\mathbf{C}$.

Now let $(\mathbf{p}^\star, c^\star) = \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$ as in Lemma C.1, with $\Sigma^\star = (\mathbf{s}_1^\star, \mathbf{s}_2^\star, \mathbf{c}_1^\star, \mathbf{c}_2^\star, \pi^\star)$. Substituting (32) into (31), if the guesses are correct, i.e., $\overline{\mathsf{id}} = \mathsf{id}_{\mathsf{ext}}^\star$ and $\overline{\tau} = \tau_{\mathsf{ext}}^\star$, then we have that

$$\mathbf{A} \cdot (\mathbf{p}_1^\star - c^\star \cdot (\mathbf{T}_1 \mathbf{s}_1^\star + \mathbf{T}_2 \mathbf{s}_2^\star)) - c^\star \cdot \mathbf{D}\mathbf{m}^\star = c^\star \cdot \mathbf{u} \bmod q. \tag{33}$$

In the following, we use the bounds $\|\mathbf{T}_1\|_2, \|\mathbf{T}_2\|_2 \leq m$, that $\|\mathbf{T}_3\mathbf{m}\|_2 \leq \sqrt{m} \cdot \ell_{\mathbf{m}}$ for any $\mathbf{m} \in \{-1, 0, 1\}^{\ell_{\mathbf{m}}}$, and that $\|\mathbf{s}_1^\star\|_2 + \|\mathbf{s}_2^\star\|_2 \leq 2\sigma_{\mathsf{sign}}\sqrt{m}$ by Verify.

Theorem C.2. *Assume that: (i) the NIZK system is sound; and (ii)* $\mathsf{SIS}_{q,n,m+1,\beta_1}$ *is hard, where* $\beta_1 = 2\alpha\sqrt{l} + C \cdot (2m\sqrt{m} \cdot \sigma_{\mathsf{sign}} + \sqrt{m} \cdot \ell_{\mathbf{m}} + 1)$. *Then the scheme in Section 4.2 is traceable against any PPT adversary* $\mathcal{A}$ *producing a type-1 forgery.*

Proof. Consider a PPT adversary $\mathcal{A}$ producing a type-1 forgery $(\mathbf{m}^\star, \Sigma^\star)$. By Lemma C.1, if $\mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star) = \perp$, then the statement constructed from $(\mathbf{m}^\star, \Sigma^\star)$ is false but the proof $\pi^\star$ verifies, and $\mathcal{A}$ directly breaks the soundness of the NIZK system. In the case that Ext outputs a pair, we construct an algorithm that solves $\mathsf{SIS}_{q,n,m+1,\beta_1}$.

Setup. The reduction $\mathcal{B}_1$ receives $(\mathbf{A} \mid \mathbf{u}) \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times (m+1)})$ as an instance of $\mathsf{SIS}_{q,n,m+1,\beta_1}$. Then $\mathcal{B}_1$ samples $\overline{\mathsf{id}} \leftarrow \mathcal{U}(S_{\mathsf{id}} \backslash [N])$ and $\overline{\tau} \leftarrow \mathcal{U}(S_\tau)$, and computes $\mathbf{B}, \mathbf{C}, \mathbf{D}$ as in (32). It generates crs as in KeyGen, keeping $(\mathbf{S}_1, \mathbf{S}_2)$, and gives gpk and $\mathsf{gmsk} = \mathbf{S}_1$ to $\mathcal{A}$.

Queries. Since $\overline{\mathsf{id}} \notin [N]$ and $N < q$, we have $\mathsf{id} \neq \overline{\mathsf{id}} \bmod q$ for *every* identity $\mathsf{id} \in [N]$. Hence $\mathcal{B}_1$ computes every user's key $\mathbf{R}_{\mathsf{id}}$ with $\mathbf{T}_1$ as described above, and answers all corruption and signing queries honestly.

Output. When $\mathcal{A}$ outputs $(\mathbf{m}^\star, \Sigma^\star)$, $\mathcal{B}_1$ computes $(\mathbf{p}^\star, c^\star) \leftarrow \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$ and outputs

$$\mathbf{w} = (\mathbf{p}_1^\star - c^\star \cdot (\mathbf{T}_1 \mathbf{s}_1^\star + \mathbf{T}_2 \mathbf{s}_2^\star + \mathbf{T}_3 \mathbf{m}^\star), -c^\star) \in \mathbb{Z}^{m+1}.$$

Analysis. The view of $\mathcal{A}$ is statistically close to that in the real experiment, and is statistically independent of $(\overline{\mathsf{id}}, \overline{\tau})$. By Lemma C.1, $\mathsf{id}_{\mathsf{ext}}^\star \in S_{\mathsf{id}} \backslash [N]$ and $\tau_{\mathsf{ext}}^\star \in S_\tau$. Thus $\overline{\mathsf{id}} = \mathsf{id}_{\mathsf{ext}}^\star$ and $\overline{\tau} = \tau_{\mathsf{ext}}^\star$ with probability at least $1/(|S_{\mathsf{id}}| \cdot |S_\tau|)$. In this case, (33) gives $(\mathbf{A} \mid \mathbf{u}) \cdot \mathbf{w} = \mathbf{0}^n \bmod q$. The vector $\mathbf{w}$ is non-zero as its last entry is $-c^\star \neq 0$, and $\|\mathbf{w}\|_2 \leq 2\alpha\sqrt{l} + C \cdot (2m\sqrt{m} \cdot \sigma_{\mathsf{sign}} + \sqrt{m} \cdot \ell_{\mathbf{m}} + 1) = \beta_1$. □

Now consider a type-2 forgery, meaning that it opens to some uncorrupted user $\mathsf{id}^\star$, and $(\mathsf{id}^\star, \mathbf{m}^\star)$ was not queried to the signing oracle. The reduction then embeds $\overline{\mathsf{id}} = \mathsf{id}^\star$, so that the $\mathbf{G}$-trapdoor $\mathbf{T}_1$ vanishes for $\mathsf{id}^\star$, and answers the signing queries of $\mathsf{id}^\star$ with the $\mathbf{G}$-trapdoor $\mathbf{T}_2$, following the strategy of [36]. We distinguish two cases, depending on the extracted tag $\tau_{\mathsf{ext}}^\star$. Let $\mathcal{T}^\star$ be the set of tags used by $\mathsf{id}^\star$ to answer signing queries. We consider two sub-cases:

– *Fresh tag:* $\tau^\star_{\mathsf{ext}} \notin \mathcal{T}^\star$. This covers any malformed tag $\tau^\star_{\mathsf{ext}} \notin [q_{\mathsf{tag}}]$, as well as any well-formed tag that $\mathsf{id}^\star$ has not used.
– *Reused tag:* $\tau^\star_{\mathsf{ext}} \in \mathcal{T}^\star$. Then $\tau^\star_{\mathsf{ext}}$ is well-formed, and it is the tag of a signing query of $\mathsf{id}^\star$ on some message $\mathbf{m}^{(+)} \neq \mathbf{m}^\star$.

Theorem C.3. *Let Q_s be a polynomial upper bound on the number of signing queries of $\mathcal{A}$. Assume that: (i) $\mathsf{SIS}_{q,n,m+1,\beta_1}$ is hard, where β_1 is as in Theorem C.2; (ii) $\mathsf{SIS}_{q,n,m,\beta_2}$ is hard, where*

$$\beta_2 = (2m+1)\cdot\Big(2\alpha\sqrt{l} + C\cdot\sigma_{\mathsf{sign}}\sqrt{2m} + C\cdot\Big(\sigma_{\mathsf{sign}}\sqrt{3m} + \sqrt{m}\cdot\ell_{\mathbf{m}}\Big)\Big) + C\cdot\sqrt{m}\cdot\ell_{\mathbf{m}}.$$

Then the scheme in Section 4.2 is traceable against any PPT adversary $\mathcal{A}$ producing a type-2 forgery.

Proof. Consider a PPT adversary $\mathcal{A}$ producing a type-2 forgery $(\mathbf{m}^\star, \Sigma^\star)$. By Lemma C.1, $(\mathbf{p}^\star, c^\star) \leftarrow \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$ satisfies $\mathsf{id}^\star_{\mathsf{ext}} = \mathsf{id}^\star$, where we have that $\mathsf{id}^\star \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$. We consider the two cases above, and follow the proof strategies of [36, Lemma 3.2, Lemma 3.3].

Reduction – Fresh Tag. The reduction $\mathcal{B}_2$ receives an instance $(\mathbf{A} \mid \mathbf{u})$ of $\mathsf{SIS}_{q,n,m+1,\beta_1}$, samples $\overline{\tau} \leftarrow \mathcal{U}(S_\tau)$ and $\mathsf{id}^\star \leftarrow \mathcal{U}([N])$, sets $\overline{\mathsf{id}} = \mathsf{id}^\star$ in (32), generates crs as in KeyGen (keeping $(\mathbf{S}_1, \mathbf{S}_2)$) and gives gpk and $\mathsf{gmsk} = \mathbf{S}_1$ to $\mathcal{A}$. Users $\mathsf{id} \neq \mathsf{id}^\star$ are handled with $\mathbf{T}_1$ as described above, since $0 < |\mathsf{id} - \mathsf{id}^\star| < N < q$. If $\mathcal{A}$ requests $\mathsf{gsk}[\mathsf{id}^\star]$, the reduction aborts. For $\mathsf{id}^\star$, note that we have

$$(\mathbf{A}_{\mathsf{id}^\star} \mid \tau\cdot\mathbf{G} - \mathbf{C}) = (\mathbf{A} \mid -\mathbf{A}\mathbf{T}_1 \mid (\tau - \overline{\tau})\cdot\mathbf{G} - \mathbf{A}\mathbf{T}_2),$$

so $(\mathbf{T}_2^\top \mid \mathbf{0})^\top \in \mathbb{Z}^{2m\times m}$ is a $\mathbf{G}$-trapdoor of this matrix with tag $(\tau - \overline{\tau})\cdot\mathbf{I}_n$, for any tag $\tau \neq \overline{\tau} \bmod q$. To answer a signing query of $\mathsf{id}^\star$ with such a tag τ, the reduction samples the JRS signature $\mathbf{x}$ with $\mathsf{SampleD}$ using this trapdoor, which has norm at most m, and performs the remaining steps of Sign honestly. The reduction $\mathcal{B}_2$ aborts whenever $\mathsf{id}^\star$ has to sign with the tag $\overline{\tau}$. When $\mathcal{A}$ outputs $(\mathbf{m}^\star, \Sigma^\star)$, $\mathcal{B}_2$ computes $(\mathbf{p}^\star, c^\star) \leftarrow \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$ and outputs $\mathbf{w}$ as in the proof of Theorem C.2.

The guesses $\mathsf{id}^\star$ and $\overline{\tau}$ are uniform over their respective domains, and hence independent of the view of $\mathcal{A}$. Thus, conditioned on non-aborting, the forgery opens to $\mathsf{id}^\star$ and $\overline{\tau} = \tau^\star_{\mathsf{ext}}$ with probability at least $1/(N\cdot|S_\tau|)$. In this case, following (33), $\mathbf{w}$ is a solution of $\mathsf{SIS}_{q,n,m+1,\beta_1}$ for $(\mathbf{A} \mid \mathbf{u})$.

Reduction – Reused Tag. The reduction $\mathcal{B}_3$ receives an instance $\mathbf{A}$ of $\mathsf{SIS}_{q,n,m,\beta_2}$. It samples $\mathsf{id}^\star \leftarrow \mathcal{U}([N])$ and $\overline{\tau} \leftarrow \mathcal{U}([Q_s])$ as the guess of the reused tag, sets $\overline{\mathsf{id}} = \mathsf{id}^\star$ in (32), generates crs as in KeyGen (keeping $(\mathbf{S}_1, \mathbf{S}_2)$) and gives gpk and $\mathsf{gmsk} = \mathbf{S}_1$ to $\mathcal{A}$. Let $\mathbf{F} = \mathbf{A}\cdot(\mathbf{I}_m \mid -\mathbf{T}_1 \mid -\mathbf{T}_2) = (\mathbf{A}_{\mathsf{id}^\star} \mid \overline{\tau}\cdot\mathbf{G} - \mathbf{C})$. Following [36], $\mathcal{B}_3$ samples $\mathbf{y} \leftarrow \mathcal{D}^{3m}_{\sigma_{\mathsf{sign}}}$ and sets $\mathbf{u} = \mathbf{F}\mathbf{y} \bmod q$. It answers the queries of users $\mathsf{id} \neq \mathsf{id}^\star$, and the signing queries of $\mathsf{id}^\star$ with tags $\tau \neq \overline{\tau}$, as in the fresh-tag case. For the $\overline{\tau}$-th query of $\mathsf{id}^\star$, on a message $\mathbf{m}^{(+)}$, it sets $\mathbf{x}^{(+)} = \mathbf{y} + (\mathbf{T}_3\mathbf{m}^{(+)}, \mathbf{0}^{2m})$, so that $\mathbf{F}\mathbf{x}^{(+)} = \mathbf{u} + \mathbf{D}\mathbf{m}^{(+)} \bmod q$, and performs the remaining steps of Sign honestly. As in the proof of [36, Lemma 3.3], since

$\left\|\mathbf{T}_3\mathbf{m}^{(+)}\right\|_2 \leq \sqrt{m}\cdot\ell_{\mathbf{m}} \leq \sigma_{\mathsf{sign}}$, the Rényi divergence of order 2 between the real signature and the simulated $\mathbf{x}^{(+)}$ is at most a constant. Therefore, if $\mathcal{A}$ outputs a type-2 forgery with a reused tag with non-negligible probability ε in the real experiment, then it does so in the reduction $\mathcal{B}_3$, with probability at least $\varepsilon^2\cdot\Theta(1)$ (conditioned on $\mathcal{B}_3$ does not abort).

When $\mathcal{A}$ outputs $(\mathbf{m}^\star, \Sigma^\star)$, $\mathcal{B}_3$ computes $(\mathbf{p}^\star, c^\star) \leftarrow \mathsf{Ext}(\mathbf{S}_1, \mathbf{m}^\star, \Sigma^\star)$. Assume that the forgery has a reused tag and the guesses are correct. Then (33) holds with $\mathbf{u} = \mathbf{F}\mathbf{x}^{(+)} - \mathbf{A}\mathbf{T}_3\mathbf{m}^{(+)} \bmod q$. Letting $\tilde{\mathbf{x}} = (\mathbf{p}_1^\star, c^\star\cdot\mathbf{s}_1^\star, c^\star\cdot\mathbf{s}_2^\star)$, this gives $\mathbf{A}\cdot\mathbf{w}' = \mathbf{0}^n \bmod q$, where

$$\mathbf{w}' = (\mathbf{I}_m \mid -\mathbf{T}_1 \mid -\mathbf{T}_2)\cdot\left(\tilde{\mathbf{x}} - c^\star\cdot\mathbf{x}^{(+)}\right) + c^\star\cdot\mathbf{T}_3\cdot\left(\mathbf{m}^{(+)} - \mathbf{m}^\star\right) \in \mathbb{Z}^m.$$

As $\mathbf{m}^{(+)} \neq \mathbf{m}^\star$ and $c^\star \neq 0$, the entropy argument of [36, Lemma 3.3] on the matrix $\mathbf{T}_3$ shows that $\mathbf{w}' \neq \mathbf{0}^m$ except with negligible probability. Moreover, $\|\tilde{\mathbf{x}}\|_2 \leq 2\alpha\sqrt{l} + C\cdot\sigma_{\mathsf{sign}}\sqrt{2m}$, $\left\|\mathbf{x}^{(+)}\right\|_2 \leq \sigma_{\mathsf{sign}}\sqrt{3m} + \sqrt{m}\cdot\ell_{\mathbf{m}}$ except with negligible probability by Lemma 2.1, and $\|(\mathbf{I}_m \mid -\mathbf{T}_1 \mid -\mathbf{T}_2)\|_2 \leq 2m+1$, so that $\|\mathbf{w}'\|_2 \leq \beta_2$. Hence $\mathbf{w}'$ is a solution of $\mathsf{SIS}_{q,n,m,\beta_2}$, and $\mathcal{B}_3$ succeeds with probability at least $\varepsilon^2/(N\cdot Q_s)\cdot\Theta(1) - \mathsf{negl}(\lambda)$. □

D Efficiency Analysis

We provide an analysis of the efficiency of the group signature scheme of Section 4.2. The parameters for evaluation include the security parameter λ and the lattice dimension $n = \Omega(\lambda)$. We treat n as an independent parameter since it specifies the dimension of the worst-case lattice problems that the construction is based on. By replacing n with λ, we get an estimation depending only on the security parameter.

Public key size The group public key gpk consists of several matrices in $\mathbb{Z}_q^{n\times m}$ with $m = \mathcal{O}(n\log q)$ and a matrix in $\mathbb{Z}_q^{n\times\ell_{\mathbf{m}}}$ where $\ell_{\mathbf{m}}$ is the maximum bit-length of the message. We assume that $\ell_{\mathbf{m}}$ is a constant, as signers can use a collision-resistant hash function to hash the messages before signing. Therefore, the bit-size of gpk is of order $\mathcal{O}(n^2\log^3\lambda)$.

Group signature size The size of a group signature is dominated by the size of the underlying NIZK, which is built upon the one-time simulation-sound NIZK system presented in Appendix B. The most-dominant part in an NIZK proof are the encryption randomnesses of the employed lossy PKE scheme. Note that, we use an $\mathcal{R}$-lossy PKE scheme to encrypt the first-move message of the trapdoor Σ-protocol of Section 3.2 that proves the language $\mathcal{L}^{\mathsf{cca}}$ specified in Proposition 4.1. In the aforementioned protocol, prover's first messages consists of a vector of $\mathbb{Z}_q^n$ and two dual-Regev ciphertexts of $\mathbb{Z}_q^{2m+k+\ell}$, where $m = \mathcal{O}(n\log q)$ and k, ℓ are of order $\mathcal{O}(\log\lambda)$. Therefore, the employed $\mathcal{R}$-lossy PKE scheme encrypts messages with a maximum bit-size of order $\mathcal{O}(n\log^2\lambda)$. This requires randomness of bit-size $\mathcal{O}(n\log^3\lambda)$, following the description of the lattice-based $\mathcal{R}$-lossy PKE scheme of [45]. Since the proof is repeated $\kappa = \Theta(\lambda/\log\lambda)$ times, the bit-size of the NIZK in a group signature is of order $\mathcal{O}(n\log^3\lambda\cdot\kappa) = \mathcal{O}(n\lambda\log^2\lambda)$.