SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2022, Vol. 65 152304:1–152304:18
https://doi.org/10.1007/s11432-020-3226-6

Lattice-based group encryptions with only one trapdoor

Jing PAN[1,2], Jiang ZHANG[2], Fangguo ZHANG[3,4], Xiaofeng CHEN[1,2*] & Willy SUSILO[5]

[1]*State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an* 710071*, China;*
[2]*State Key Laboratory of Cryptology, P.O. Box 5159, Beijing* 100878*, China;*
[3]*School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou* 510006*, China;*
[4]*Guangdong Key Laboratory of Information Security, Guangzhou* 510006*, China;*
[5]*Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong* NSW 2522*, Australia*

Received 24 October 2020/Revised 21 December 2020/Accepted 1 March 2021/Published online 16 March 2022

Abstract Group encryption (GE), the encryption analog of group signatures, is a fundamental primitive that offers a privacy-preserving service for a specific receiver concealed within a group of certified users. Like other cryptographic primitives, GE constructions are always considered relative to the potential danger of quantum computations. The only existing lattice-based variant appeared in the work of Libert et al. (Asiacrypt'16). Despite its non-trivial achievement, the construction suffers in terms of efficiency due to the extensive use of lattice trapdoors. In this paper, we develop an integrated zero-knowledge argument system that is friendly to both accumulated values and hidden matrices and supports efficient designs from lattices. Based on this system, we propose efficiency enhancing GE where only group users are required to possess the lattice trapdoors and the other parties are not. In particular, we utilize lattice-based cryptographic accumulators to confirm prospective group members and use the dual Regev encryption scheme to provide privacy for ciphertext recipients. These modifications significantly increase GE efficiency. In addition, under the intractability assumptions of the standard lattice problems, we prove the security of the proposed scheme in the standard model (assuming interaction during the proof phase), which retains the strongest level of security as the only currently available candidate.

Keywords lattice cryptography, group encryptions, lattice trapdoors, accumulators, zero-knowledge

Citation Pan J, Zhang J, Zhang F G, et al. Lattice-based group encryptions with only one trapdoor. Sci China Inf Sci, 2022, 65(5): 152304, https://doi.org/10.1007/s11432-020-3226-6

1 Introduction

Group encryption (GE) is a fundamental anonymity primitive introduced by Kiayias, Tsiounis, and Yung (KTY) [1] to protect the identities of valid users who are allowed to decrypt the well-formed ciphertexts. In general, GE is known as the natural encryption analog of group signatures [2] that, in a similar manner, conceal honest signers within a certified population. Except for slight differences, GEs and group signatures share common design concepts and have similar structures. For salient properties, such as ciphertext verifiability and user identity privacy, GE is widely applied to intercepting uncertified encrypted emails, building oblivious retriever storage systems, and hierarchical group signatures [3].

In practice, the most frequently considered goals of GE schemes are security and efficiency. Libert et al. [4] proposed the first lattice-based construction under the classical assumptions of the standard learning with errors (LWE) [5] and the short integer solutions (SIS) [6] to withstand quantum attacks. The proposed construction realizes all the specific functionalities defined in [1] via ordinary digital signatures [7] and Agrawal-Boneh-Boyen (ABB) encryption variants [4], both of which rely heavily on lattice trapdoors. However, inefficiency was inevitable; the use of lattice trapdoors implies a significant gap between

* Corresponding author (email: xfchen@xidian.edu.cn)

theoretical design and practical implementation. The analysis provided in a previous study [8] indicates that lattice trapdoors, such as Gentry-Peikert-Vaikuntanathan (GPV) trapdoors [9] and Micciancio-Peikert trapdoors [10, 11], can help construct the most ubiquitous lattice-based cryptographic schemes. However, designing efficient cryptographic schemes using the currently best trapdoor-generation algorithms [6, 10, 12] is difficult, since the allowed parameters are commonly large.

Eliminating the use of trapdoors is a reliable way to deal with the efficiency issue. Actually, it improves the practicality of lattice-based schemes in essential. Following this idea, we consider taking several modification strategies for the scheme [4]: We use lattice-based cryptographic accumulators [13] rather than the lattice-trapdoor-based digital signatures [7] to verify group membership. In addition, we replace ABB encryption variants [4] with the double LWE encryption mechanism [13, 14] to protect anonymous recipients. To some extent, these modifications reduce the extreme reliance on lattice trapdoors, significantly increase the range from which system parameters can be selected, and provide a more efficient scheme.

Our contributions. Motivated by the dramatic efficiency of lattice-based cryptographic schemes that do not involve trapdoors, we start with the group encryption [4] that involves multiple lattice trapdoors [9, 15], and employ the techniques shown in [13] to build a variant that only requires a single trapdoor. Our primary contributions can be summarized as follows:

• We develop a zero-knowledge argument system that is friendly to both the lattice-based accumulators and the quadratic relation of a matrix-vector with a hidden matrix. It supports the construction of cryptographic schemes that rely on fewer or even no lattice trapdoors and is further helpful for addressing the efficiency problem associated with the use of trapdoors.

• Building on the above zero-knowledge argument system and the dual Regev encryption mechanism, we realize a much more efficient lattice-based group encryption scheme by using fewer trapdoors. Compared to the strongest security level of the only currently existing scheme [4], our scheme obtains drastically shorter keys and ciphertexts and lower communication cost; however, as yet it is unpractical.

Related work. The group encryption primitive was first formalized by Kiayias et al. [1]. They offered a helpful design routine by simultaneously applying zero-knowledge (ZK) proofs, appropriate digital signatures (e.g., [16]), and anonymous CCA2-secure public-key encryptions (e.g., [17]). Later, Cathalo et al. [18] improved the initially interactive scheme by developing a non-interactive case using a fresh public key certification scheme and standard techniques that incurred the smallest amount of interaction. Working toward a practical implementation, Aimani et al. [19] utilized succinct approaches to hide the identities of group members over weaker assumptions. To better balance privacy vs. safety, Libert et al. [20] proposed a variant with public traceability to specific ciphertexts that shared functionality similar to that of traceable signatures [21]. Further, Izabachène et al. [22] constructed traceable group encryptions free of subliminal channels, stressing confidentiality, anonymity, and traceability.

All the group encryptions described above are vulnerable to quantum computers in that they were proposed based on the number-theoretic hardness, e.g., factoring or computing composite degree residuosity classes. To address this, Libert et al. [4] proposed the only currently existing lattice-based construction via extensive use of lattice trapdoors throughout the design, which made the instantiation suffer low efficiency and significantly worse parameter choices. Then, to solve the inefficiency problem, we employ several trapdoor-free cryptographic techniques to construct an efficiency enhancing group encryption that achieves CCA2-secure encryption [23] and anonymous recipients [1] over the same lattice assumptions.

Organization. In Section 2, we introduce the associated concepts and definitions of group encryptions and lattice-based cryptography, and recall the Libert-Ling-Nguyen-Wang (LLNW) lattice-based accumulators. Section 3 develops an integrated Stern-like argument system, which is friendly to accumulated values and hidden matrices, and serves as the underlying protocol for our construction. Our main scheme is described and analyzed in Section 4. Conclusion and suggestions for future work are provided in Section 5.

2 Preliminaries

In this section, initially we provide the formal notions and definitions of the group encryption primitive, and the computationally hard lattice problems to be believed. Then, lattice-based accumulators, which are used to verify the group membership, are recalled.

2.1 Group encryptions

We adopt the syntax and security model similar to that of [1] except for some differences[1)], where the GE primitive involves several parties (i.e., a sender, a verifier, a group manager (GM) issuing identifiers to recipients of ciphertexts and an opening authority (OA) who reveals recipients' identities), and is specified by the following algorithms and protocols.

• $\mathsf{SETUP}(\lambda)$: This algorithm consists of the following three procedures and generates group public key $\mathsf{gpk} = (\mathsf{pp}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}})$ as follows:

(1) $\mathsf{SETUP}_{\mathsf{init}}(1^{\lambda})$: Given the security parameter λ, the procedure samples public parameters pp.

(2) $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{pp})$: Given parameters pp, the procedure returns a key pair $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$ for the GM.

(3) $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{pp})$: On input pp, the procedure outputs a key pair $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the OA.

An interaction is occurred between the GM and the OA, successfully creating the group public key gpk and initializing the registration table **reg** at its end.

• $\mathsf{UKGEN}(\mathsf{pp})$: Given parameters pp, this algorithm returns the user a key pair $(\mathsf{pk}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}})$.

• $\langle \mathsf{JOIN}(\mathsf{gpk}, \mathsf{pk}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}}), \mathsf{ISSUE}(\mathsf{sk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{U}}) \rangle$: This is an interaction run by the GM and a prospective user, whose successful completion joins the new group member with a membership identifier $\mathsf{cert}_{\mathsf{U}}$.

• $\langle \mathcal{G}_r, \mathsf{sample}_{\mathcal{R}} \rangle(\mathsf{pp}, \mathcal{R})$: Given parameters pp and relation $\mathcal{R}$, procedure $\mathcal{G}_r$ produces a key pair $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$, which allows sampler $\mathsf{sample}_{\mathcal{R}}$ to generate a pair $(x, w) \in \mathcal{R}$.

• $\mathsf{ENC}(\mathsf{gpk}, \mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}}, M)$: This algorithm is executed by the sender to compute a group encryption Ψ on message M under some public key pk_{U}.

• $\mathsf{DEC}(\mathsf{sk}_{\mathsf{U}}, \Psi, L)$: The intended receiver decrypts the ciphertext Ψ.

• $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}, \mathbf{reg}, \Psi, L)$: This algorithm is run by the OA to return an identity U of a group member or $\perp$ if it fails to trace the receiver.

• $\langle \mathcal{P}(\mathsf{gpk}, \mathcal{R}, \mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}}, \Psi, \mathrm{coins}_{\Psi}), \mathcal{V}(\mathsf{gpk}, \Psi, \pi_{\Psi}) \rangle$: The sender and verifier carry out the interactive procedure between them, given inputs, to convince the verifier that the ciphertext Ψ is actually generated for one of group members.

For the security requirements of our GE scheme, as in [1], correctness, message secrecy, anonymity, and soundness are considered and analyzed. Here we only give the informal statements, and their formal definitions are referred to [1].

Correctness asks that a ciphertext generated by a genuine sender is always decrypted successfully by decryptor DEC, and that procedure OPEN is always able to correctly identify the recipient, while producing an accepted proof.

Message secrecy demands that, for any probabilistic polynomial-time (PPT) adversary, it is difficult to distinguish a random ciphertext from a one generated under a specific relation, even if it can corrupt all parties except the honest receiver, can pick the GM's key, and is allowed access to the OPEN and PROVE oracles.

Anonymity says that, for any PPT adversary, it is impossible to distinguish ciphertexts computed under two valid public keys it chooses, even when it controls the entire system except OA, and is allowed for the OPEN oracle.

Soundness requires that, for any PPT adversary, it is quite difficult to produce a convincing valid ciphertext that can be opened to an unregistered group member or an invalid public key, even if it can choose OA's key, and is granted access to the REG oracle.

2.2 Computational lattice problems

The security of our scheme relies on the hardness of the following standard lattice problems, which are believed difficult to solve in polynomial time.

Definition 1 (SIS)**.** Given parameter n and appropriate positive integers m, q, β, the $\mathsf{SIS}_{n,m,q,\beta}$ demands, for any $\boldsymbol{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, to search a vector $\boldsymbol{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ with norm bound β such that $\boldsymbol{A} \cdot \boldsymbol{x} = \mathbf{0}$.

By choosing appropriate parameters, the worst-case lattice problem SIVP_{γ} can be reduced to the average-case one $\mathsf{SIS}_{n,m,q,\beta}$. Such an example follows by setting $m, \beta = \mathsf{poly}(n)$; $q \geqslant \sqrt{n}\beta$ and $\gamma = \widetilde{\mathcal{O}}(\sqrt{n}\beta)$ (e.g., [6, 9, 24]).

1) The KTY model is defined for dynamic enrollments, which allows the population to grow. Our model does not involve dynamic enrollments; the group is fixed once all potential users have joined. Although our model is somewhat less dynamic compared to [4], it is still of interest for use in some realistic scenarios and in terms of efficiency gain.

Definition 2 (LWE)**.** Given appropriate positive integers n, m, q, and a probability distribution on $\mathbb{Z}$ denoted as χ, for secret $\boldsymbol{s} \in \mathbb{Z}_q^n$, define $\boldsymbol{A}_{\boldsymbol{s},\chi}$ as the distribution generated by sampling $\boldsymbol{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and $e \leftarrow \chi$, and returning $(\boldsymbol{a}, \boldsymbol{a}^{\mathrm{T}} \cdot \boldsymbol{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The goal of $\mathsf{LWE}_{n,q,\chi}$ is to distinguish m samples from $\boldsymbol{A}_{\boldsymbol{s},\chi}$ and m samples from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$, respectively.

For prime power q, one can build a discrete integer distribution χ bounded by $B \geqslant \sqrt{n}\omega(\log n)$, for which there exists an efficient reduction from the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(nq/B)}$ problem to the $\mathsf{LWE}_{n,q,\chi}$ problem (e.g., [5, 25, 26]).

2.3 LLNW lattice-based Merkle-tree accumulators

Our construction takes the LLNW lattice-based accumulator [13] as a crucial building block to enroll the prospective users into the group in a free-of-trapdoors manner. The accumulator shares the same properties as the number-theoretic counterparts [27–30], and is built on the family of hash functions $\mathcal{H} = \{h_{\boldsymbol{A}} | \boldsymbol{A} \in \mathbb{Z}_q^{n \times m}\}$ that map any input pair $(\boldsymbol{u}_0, \boldsymbol{u}_1) \in (\{0,1\}^{nk})^2$ to $h_{\boldsymbol{A}}(\boldsymbol{u}_0, \boldsymbol{u}_1) = \mathsf{bin}(\boldsymbol{A}_0 \cdot \boldsymbol{u}_0 + \boldsymbol{A}_1 \cdot \boldsymbol{u}_1 \bmod q) \in \{0,1\}^{nk}$ and have the collision resistance property based on the SIS problem. Additionally, the accumulator supports the zero-knowledge argument of knowledge (ZKAoK). Here, we set integers $k = \lceil \log q \rceil, m = 2nk$ and take the matrix $\boldsymbol{A} = [\boldsymbol{A}_0 | \boldsymbol{A}_1] \in \mathbb{Z}_q^{n \times m}$ consisting of two same-size blocks, and yet use the notation $\mathsf{bin}(\cdot)$ to represent the binary decomposition function.

Informally, the accumulator is described via a tuple of algorithms $(\mathsf{TSetup}, \mathsf{TAcc}, \mathsf{TWitness}, \mathsf{TVerify})$. Namely, given a Merkle-tree with $N = 2^\ell$ leaves, algorithm TSetup samples a random matrix $\boldsymbol{A}$ for hash function $h_{\boldsymbol{A}}$; algorithm TAcc accumulates all given values $R = \{\boldsymbol{d}_0 \in \{0,1\}^{nk}, \ldots, \boldsymbol{d}_{N-1} \in \{0,1\}^{nk}\}$ on leaves into the root $\boldsymbol{u}$ via a recursive computation $\boldsymbol{u}_{b_1,\ldots,b_i} = h_{\boldsymbol{A}}(\boldsymbol{u}_{b_1,\ldots,b_i,0}, \boldsymbol{u}_{b_1,\ldots,b_i,1})$ for any node at depth $i \in [\ell]$ with $(b_1, \ldots, b_i) \in \{0,1\}^i$, given the initial definition $\boldsymbol{u} = h_{\boldsymbol{A}}(\boldsymbol{u}_0, \boldsymbol{u}_1)$ for the root $\boldsymbol{u}$; and algorithm TWitness returns $\perp$ if $\boldsymbol{d} \notin R$ or the witness $w = ((j_1, \ldots, j_\ell), (\boldsymbol{u}_{j_1,\ldots,j_{\ell-1},\bar{j}_\ell}, \ldots, \boldsymbol{u}_{j_1,\bar{j}_2}, \boldsymbol{u}_{\bar{j}_1})) \in \{0,1\}^\ell \times (\{0,1\}^{nk})^\ell$ that demonstrates that $\boldsymbol{d} \in R$ for some $j \in [0, N-1]$ with binary form $(j_1, \ldots, j_\ell)$ such that $\boldsymbol{d} = \boldsymbol{d}_j$, where $\bar{b}$ denotes the bit $1-b$ for any bit b; finally, given witness $w = ((j_1, \ldots, j_\ell), (\boldsymbol{w}_\ell, \ldots, \boldsymbol{w}_1)) \in \{0,1\}^\ell \times (\{0,1\}^{nk})^\ell$, and set $\boldsymbol{v}_\ell = \boldsymbol{d}$, algorithm TVerify computes the path $\boldsymbol{v}_{\ell-1}, \ldots, \boldsymbol{v}_0 \in \{0,1\}^{nk}$ in the recursive fashion by using the formula $\boldsymbol{v}_i = \bar{j}_{i+1} \cdot h_{\boldsymbol{A}}(\boldsymbol{v}_{i+1}, \boldsymbol{w}_{i+1}) + j_{i+1} \cdot h_{\boldsymbol{A}}(\boldsymbol{w}_{i+1}, \boldsymbol{v}_{i+1})$ for any $j \in [0, N-1]$ and $i \in [\ell - 1]$ with initial setting $\boldsymbol{u} = \boldsymbol{v}_0$. It is proved that the accumulator is secure under the infeasibility assumption of the $\mathsf{SIS}_{n,m,q,1}$ problem.

3 The supporting zero-knowledge argument system

We first recall the decomposition, extension, and permutation techniques applied in [4, 13] and build several sophisticated permutation techniques (special permutations $F_{\boldsymbol{b},\pi}^{(t)}(\cdot)$ and $F_{\pi}^{(t)}(\cdot)$). Then, we build our statistical ZKAoK that plays a crucial role in our GE scheme.

Before this section, it should be first noted that, as in [4], our argument system is also Stern-type statical zero-knowledge [31] that can produce a simulated transcript having a negligibly statistical distance to that produced from the interaction carried out by honest prover and any verifier, and is further Σ-protocols in the generalized sense [32, 33] where extraction needs 3 instead of just 2 valid transcripts. More recently, numerous cryptographic constructions based on lattice problems [34–36] or code problems [33] are designed with the help of these protocols.

3.1 Decompositions, extensions, and permutations

Decompositions. For any integer number $B \in \mathbb{Z}_+$, by setting $\delta_B := \lfloor \log_2 B \rfloor + 1$ and computing the sequence $\{B_1, B_2, \ldots, B_{\delta_B}\}$ via $B_j = \lfloor \frac{B + 2^{j-1}}{2^j} \rfloor$, $\forall j \in [1, \delta_B]$, we can decompose any integer $i \in [0, B]$ into $i = \sum_{j=1}^{\delta_B} i_j \cdot B_j$, which directly gives a desired bit vector as $\mathsf{idec}_B(i) = (i_1, \ldots, i_{\delta_B})^{\mathrm{T}} \in \{0,1\}^{\delta_B}$. The decomposition method can be executed in a deterministic manner as shown in [4], and can further be adapted to decomposing vectors and matrices after combining with the matrix $\boldsymbol{H}_{\mathfrak{m},B} = \boldsymbol{I}_{\mathfrak{m}} \otimes (B_1 \ B_2 \ \cdots \ B_{\delta_B}) \in \mathbb{Z}_q^{\mathfrak{m} \times \mathfrak{m}\delta_B}$, as follows:

– $\mathsf{vdec}_{\mathfrak{m},B}$: decomposes $\mathfrak{m}$-size vector $\boldsymbol{v} = (v_1, \ldots, v_{\mathfrak{m}})^{\mathrm{T}}$ with $v_i \in [0, B]$ for each $i = 1, \ldots, \mathfrak{m}$, into $(\mathsf{idec}_B(v_1)^{\mathrm{T}} \| \cdots \| \mathsf{idec}_B(v_{\mathfrak{m}})^{\mathrm{T}})^{\mathrm{T}} \in \{0,1\}^{\mathfrak{m}\delta_B}$, holding that $\boldsymbol{v} = \boldsymbol{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}_{\mathfrak{m},B}(\boldsymbol{v})$.

– $\mathsf{vdec}'_{\mathfrak{m},B}$: maps any vector $\boldsymbol{w}=(w_1,\ldots,w_{\mathfrak{m}})^{\mathrm{T}}\in[-B,B]^{\mathfrak{m}}$ to the vector $(\sigma(w_1)\cdot\mathsf{idec}_B(|w_1|)^{\mathrm{T}}\|\cdots\|\sigma(w_{\mathfrak{m}})\cdot\mathsf{idec}_B(|w_{\mathfrak{m}}|)^{\mathrm{T}})^{\mathrm{T}}\in\{-1,0,1\}^{\mathfrak{m}\delta_B}$, where $\sigma(\cdot)$ is the sign$(\cdot)$ function which outputs -1, 0 or 1 according to negative, 0 or positive inputs, respectively. Similarly, this gives that $\boldsymbol{w}=\boldsymbol{H}_{\mathfrak{m},B}\cdot\mathsf{vdec}'_{\mathfrak{m},B}(\boldsymbol{w})$.

– $\mathsf{mdec}_{n,m,q}$: decomposes a given matrix $\boldsymbol{X}=[\boldsymbol{x}_1|\cdots|\boldsymbol{x}_n]\in\mathbb{Z}_q^{m\times n}$ with $\{\boldsymbol{x}_i\}_i^n\in\mathbb{Z}_q^m$, to a binary vector $(\mathsf{vedc}_{m,q-1}(\boldsymbol{x}_1)^{\mathrm{T}}\|\cdots\|\mathsf{vedc}_{m,q-1}(\boldsymbol{x}_n)^{\mathrm{T}})^{\mathrm{T}}\in\{0,1\}^{nm\delta_{q-1}}$. In the framework of Stern-type zero-knowledge argument system, the generalized matrix decomposition is applied in proving knowledge of a lattice relation with hidden matrices, and we defer the involved statement to the following discussion of expansion operator $\mathsf{expand}^{\otimes}(\cdot,\cdot)$.

Extensions. For simplicity, we use B_{2t}^t to denote the set of all length-$2t$ binary vectors with exact Hamming weight t, and use B_{3t}^t to represent the set of length-$3t$ vectors that share the equal t's number for each entry of $\{-1,0,1\}$, then start with the simple encoding techniques.

– $\mathsf{enc}(\cdot,\cdot)$: $\{0,1\}\times\{0,1\}^m\to\{0,1\}^{2m}$ extends a size-m vector bit $\boldsymbol{v}$ to a vector of the form $\boldsymbol{z}=\left(\begin{smallmatrix}\bar{b}\,\vdots\,\boldsymbol{v}\\ b\,\vdots\,\boldsymbol{v}\end{smallmatrix}\right)$ for a choice of bit b.

– $\mathsf{enc}_t(\cdot)$: $\{0,1\}^t\to\{0,1\}^{2t}$ maps any bit vector $\boldsymbol{x}=(x_1,\ldots,x_t)^{\mathrm{T}}$ to a double-size bit vector of form $\mathsf{enc}_t(\boldsymbol{x})=(\bar{x}_1,x_1,\ldots,\bar{x}_t,x_t)^{\mathrm{T}}\in\{0,1\}^{2t}$. Given a random bit x, a trivial example is shown as $\mathsf{enc}_1(x)=(\bar{x},x)^{\mathrm{T}}$.

– $\mathsf{ext}_2(\cdot)$: $\{0,1\}^t\to\mathsf{B}_{2t}^t$. Extend length-$t$ bit vector $\boldsymbol{x}$ with Hamming weight hw to the double-length vector $\boldsymbol{x}'=[\boldsymbol{x}^{\mathrm{T}}\|(\mathbf{1}^{(t-hw)})^{\mathrm{T}}\|(\mathbf{0}^{(hw)})^{\mathrm{T}}]^{\mathrm{T}}$.

– $\mathsf{ext}_3(\cdot)$: $\{0,1\}^t\to\mathsf{B}_{3t}^t$. Getting $\boldsymbol{y}$ with exact n_i entries i, $\forall i\in\{-1,0,1\}$ as input, it gives $\boldsymbol{y}'=[\boldsymbol{y}^{\mathrm{T}}\|((-\mathbf{1})^{(t-n_{-1})})^{\mathrm{T}}\|(\mathbf{0}^{(t-n_0)})^{\mathrm{T}}\|(\mathbf{1}^{(t-n_1)})^{\mathrm{T}}]^{\mathrm{T}}$. This function is mainly used in extending the decompositions of vectors with infinity norms bounded by some integer B (see, e.g., [4,7]).

– $\mathsf{Ext}(\cdot,\cdot)$: $\{0,1\}^2\to\{0,1\}^4$ encodes a bit pair $(e_1,e_2)\in\{0,1\}^2$ to a four-bit multiplication vector $(\bar{e}_1\cdot\bar{e}_2,\bar{e}_1\cdot e_2,e_1\cdot\bar{e}_2,e_1\cdot e_2)^{\mathrm{T}}$.

– $\mathsf{expand}^{\otimes}(\cdot,\cdot)$: $\{0,1\}^{nmk}\times\{0,1\}^{nk}\to\{0,1\}^{4nmk^2}$. By imposing the function $\mathsf{Ext}(\cdot,\cdot)$ on all possible pairs of the form $(x_{i,j},s_{i,t})$, where the index i, j, t runs through $[n]$, $[mk]$, $[k]$, respectively, this operator maps two bit vectors $\boldsymbol{X}_0=(x_{1,1},\ldots,x_{1,mk},\ldots,x_{i,1},\ldots,x_{i,mk},\ldots,x_{n,1},\ldots,x_{n,mk})^{\mathrm{T}}$ and $\boldsymbol{s}_0=(s_{1,1},\ldots,s_{1,k},\ldots,s_{i,1},\ldots,s_{i,k},\ldots,s_{n,1},\ldots,s_{n,k})^{\mathrm{T}}$ to vector $\boldsymbol{z}$ of the form

$$\begin{aligned}&\big(\mathsf{Ext}^{\mathrm{T}}(x_{1,1},s_{1,1})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{1,1},s_{1,t})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{1,1},s_{1,k})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{i,j},s_{i,1})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{i,j},s_{i,t})\|\\&\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{i,j},s_{i,k})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{n,mk},s_{n,1})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{n,mk},s_{n,t})\|\cdots\|\mathsf{Ext}^{\mathrm{T}}(x_{n,mk},s_{n,k})\big)^{\mathrm{T}},\end{aligned}$$

which defines $\boldsymbol{z}=\mathsf{expand}^{\otimes}(\boldsymbol{X}_0,\boldsymbol{s}_0)$.

We note that the last above extension $\mathsf{expand}^{\otimes}(\cdot,\cdot)$ is crucial in representing the matrix-vector product for hidden matrices. By combining with the above decompositions for vectors and matrices, we can equally write the matrix-vector product with hidden matrices as the matrix-vector product where the matrices are public. Namely, let $\{q_1,\ldots,q_k\}$ be the sequence of integers computed by function $\mathsf{idec}_{q-1}(\cdot)$, set $\boldsymbol{g}=(q_1,\ldots,q_k)$ and $\boldsymbol{g}'=(0,0,0,q_1,\ldots,0,0,0,q_k)\in\mathbb{Z}_q^{4k}$, then for $\boldsymbol{X}\in\mathbb{Z}_q^{m\times n}$ and $\boldsymbol{s}\in\mathbb{Z}_q^n$, one can obtain

$$\boldsymbol{X}\cdot\boldsymbol{s}\bmod q=\boldsymbol{Q}\cdot\mathsf{expand}^{\otimes}\big(\boldsymbol{X}_0,\boldsymbol{s}_0\big)\bmod q,$$

where $\boldsymbol{X}_0=\mathsf{mdec}_{n,m,q}(\boldsymbol{X})$, $\boldsymbol{s}_0=\mathsf{vdec}_{n,q-1}(\boldsymbol{s})$, and $\boldsymbol{Q}=\boldsymbol{H}_{m,q-1}\cdot\widehat{\boldsymbol{Q}}\in\mathbb{Z}_q^{m\times 4nmk^2}$, $\widehat{\boldsymbol{Q}}=[\overbrace{\boldsymbol{Q}_0|\cdots|\boldsymbol{Q}_0}^{n\text{ times}}]\in\mathbb{Z}_q^{mk\times 4nmk^2}$, and $\boldsymbol{Q}_0:=\boldsymbol{I}_{mk}\otimes\boldsymbol{g}'\in\mathbb{Z}_q^{mk\times 4mk^2}$. This equally allows writing the equation $\boldsymbol{b}=\boldsymbol{X}\cdot\boldsymbol{s}+\boldsymbol{e}\bmod q$ with the secret input $(\boldsymbol{X},\boldsymbol{s},\boldsymbol{e})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^n\times[-\beta,\beta]^m$ as the common equation $\boldsymbol{b}=\boldsymbol{Q}\cdot\boldsymbol{z}+\boldsymbol{H}_{m,\beta}\cdot\boldsymbol{e}_0\bmod q$, where the secret vectors are given by $\boldsymbol{z}=\mathsf{expand}^{\otimes}(\boldsymbol{X}_0,\boldsymbol{s}_0)\in\{0,1\}^{4nmk^2}$, $\boldsymbol{X}_0=\mathsf{mdec}_{n,m,q}(\boldsymbol{X})\in\{0,1\}^{nmk}$, $\boldsymbol{s}_0=\mathsf{vdec}_{n,q-1}(\boldsymbol{s})\in\{0,1\}^{nk}$, and $\boldsymbol{e}_0=\mathsf{vdec}'_{m,\beta}(\boldsymbol{e})\in\{-1,0,1\}^{m\delta_\beta}$.

Permutations. Before the following statement, it is first noted that we abuse the notation of transposition in the first three definitions, and denote by $\mathcal{S}_i$ the symmetric group consisting of all possible permutations for i elements. The involved permutations are given as follows.

– $T_{\boldsymbol{b}}^{(t)}(\cdot):(\mathbb{Z}_q^m)^{2t}\to(\mathbb{Z}_q^m)^{2t}$. For any vector $\boldsymbol{b}=(b_1,\ldots,b_t)^{\mathrm{T}}\in\{0,1\}^t$, it transforms a vector $\boldsymbol{z}=(\boldsymbol{z}_1^{(0)},\boldsymbol{z}_1^{(1)},\ldots,\boldsymbol{z}_t^{(0)},\boldsymbol{z}_t^{(1)})^{\mathrm{T}}$ into the vector $T_{\boldsymbol{b}}^{(t)}(\boldsymbol{z})=(\boldsymbol{z}_1^{(b_1)},\boldsymbol{z}_1^{(\bar{b}_1)},\ldots,\boldsymbol{z}_t^{(b_t)},\boldsymbol{z}_t^{(\bar{b}_t)})^{\mathrm{T}}$. As a special case, set $\boldsymbol{x}=(x_1,\ldots,x_t)^{\mathrm{T}}\in\{0,1\}^t$ and $\boldsymbol{y}=\mathsf{enc}_t(\boldsymbol{x})$, this holds the following equivalence:

$$\boldsymbol{y}=\mathsf{enc}_t(\boldsymbol{x})\Longleftrightarrow T_{\boldsymbol{b}}^{(t)}(\boldsymbol{y})=\mathsf{enc}_t(\boldsymbol{x}\oplus\boldsymbol{b}).\tag{1}$$

– $F^{(t)}_{\boldsymbol{b},\pi}(\cdot) : (\mathbb{Z}_q^m)^{2t} \to (\mathbb{Z}_q^m)^{2t}$. Given a random bit vector $\boldsymbol{b} = (b_1, \ldots, b_t)^{\mathrm{T}} \in \{0,1\}^t$ and a set of permutations $\pi = (\pi_1, \ldots, \pi_t)$, it transforms a vector $\boldsymbol{z} = (\boldsymbol{z}_1^{(0)}, \boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_t^{(0)}, \boldsymbol{z}_t^{(1)})^{\mathrm{T}}$ into the vector $F^{(t)}_{\boldsymbol{b},\pi}(\boldsymbol{z}) = (\pi_1(\boldsymbol{z}_1^{(b_1)}), \pi_1(\boldsymbol{z}_1^{(\bar{b}_1)}), \ldots, \pi_t(\boldsymbol{z}_t^{(b_t)}), \pi_t(\boldsymbol{z}_t^{(\bar{b}_t)}))^{\mathrm{T}}$. Namely, $F^{(t)}_{\boldsymbol{b},\pi}$ first arranges the each block of the targeted vector $\boldsymbol{z}$ according to bit vector $\boldsymbol{b}$, then permutes the corresponding block under π_i for each $i \in [t]$. A trivial example is given by $F^{(1)}_{b,\pi}(\boldsymbol{z}) = (\pi(\boldsymbol{z}_b^{\mathrm{T}}), \pi(\boldsymbol{z}_{\bar{b}}^{\mathrm{T}}))^{\mathrm{T}}$ under taking $t = 1, b \in \{0,1\}$ and a permutation π. As a special case, set $\boldsymbol{z} = (\mathsf{enc}^{\mathrm{T}}(c_1, \boldsymbol{v}_1), \ldots, \mathsf{enc}^{\mathrm{T}}(c_t, \boldsymbol{v}_t))$ where vectors $\boldsymbol{v}_i \in \mathsf{B}^t_{2t}$ and $\pi_i \in \mathcal{S}_{2t}$, $\forall i \in [t]$, we obtain that: $F^{(t)}_{\boldsymbol{b},\pi}(\boldsymbol{z}) = (\mathsf{enc}^{\mathrm{T}}(c_1 \oplus b_1, \pi_1(\boldsymbol{v}_1)), \ldots, \mathsf{enc}^{\mathrm{T}}(c_t \oplus b_t, \pi_t(\boldsymbol{v}_t)))^{\mathrm{T}}$. The result is applicable in the construction of our ZKAoK system.

– $F^{(t)}_{\pi}(\cdot) : (\mathbb{Z}_q^{m_1} \times \cdots \times \mathbb{Z}_q^{m_t}) \to (\mathbb{Z}_q^{m_1} \times \cdots \times \mathbb{Z}_q^{m_t})$. Given a set of permutations $\pi = (\pi_1, \ldots, \pi_t)$, it transforms a vector $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t)^{\mathrm{T}}$ into the vector $F^{(t)}_{\pi}(\boldsymbol{z}) = (\pi_1(\boldsymbol{z}_1), \ldots, \pi_t(\boldsymbol{z}_t))^{\mathrm{T}}$. Namely, $F^{(t)}_{\pi}$ arranges each block of $\boldsymbol{z}$ according to the corresponding entry of π for each $i \in [t]$.

– $T_{b_1,b_2}(\cdot) : \mathbb{Z}^4 \to \mathbb{Z}^4$, where $b_1, b_2 \in \{0,1\}$. It transforms any integer vector of the form $\boldsymbol{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^{\mathrm{T}} \in \mathbb{Z}_q^4$ into the vector of the form $(v_{b_1,b_2}, v_{b_1,\bar{b}_2}, v_{\bar{b}_1,b_2}, v_{\bar{b}_1,\bar{b}_2})^{\mathrm{T}}$, which supports the following equation:

$$\boldsymbol{z} = \mathsf{Ext}(c_1, c_2) \Longleftrightarrow T_{b_1,b_2}(\boldsymbol{z}) = \mathsf{Ext}(c_1 \oplus b_1, c_2 \oplus b_2). \tag{2}$$

– $P_{\boldsymbol{a},\boldsymbol{b}}(\cdot) : \mathbb{Z}^{4nmk^2} \to \mathbb{Z}^{4nmk^2}$. Given the length-$nmk$ binary vector $\boldsymbol{a}$ and the length-nk binary vector $\boldsymbol{b}$ consisting of $\{a_{i,j}\}_{i,j}$ and $\{b_{i,t}\}_{i,t}$, respectively, it transforms an integer vector $\boldsymbol{v} \in \mathbb{Z}_q^{4nmk^2}$ consisting of nmk^2 vectors $\boldsymbol{v}_{i,j,t}$ of the form $(\boldsymbol{v}_{i,j,t}^{(0,0)}, \boldsymbol{v}_{i,j,t}^{(0,1)}, \boldsymbol{v}_{i,j,t}^{(1,0)}, \boldsymbol{v}_{i,j,t}^{(1,1)})^{\mathrm{T}}$ according the lexicographic order for all $(i,j,t) \in [n] \times [mk] \times [k]$, to the same-size vector $\boldsymbol{w}$ consisting of $\boldsymbol{w}_{i,j,t}$ via the basic transformation $\boldsymbol{w}_{i,j,t} = T_{a_{i,j},b_{i,t}}(\boldsymbol{v}_{i,j,t})$. This leads the following equivalence:

$$\boldsymbol{z} = \mathsf{expand}^{\otimes}(\boldsymbol{X}_0, \boldsymbol{s}_0) \Longleftrightarrow P_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{z}) = \mathsf{expand}^{\otimes}(\boldsymbol{X}_0 \oplus \boldsymbol{a}, \boldsymbol{s}_0 \oplus \boldsymbol{b}). \tag{3}$$

It is remarked that the above permutation $F^{(t)}_{\boldsymbol{b},\pi}(\cdot)$ is crucial for proving accumulated values in the ZK manner. Actually, the key component of the proof is what concerns the algorithm $\mathsf{TVerify}(\cdot)$ given by

$$\begin{aligned} &\boldsymbol{v}_i = \bar{j}_{i+1} \cdot h_{\boldsymbol{A}}(\boldsymbol{v}_{i+1}, \boldsymbol{w}_{i+1}) + j_{i+1} \cdot h_{\boldsymbol{A}}(\boldsymbol{w}_{i+1}, \boldsymbol{v}_{i+1}) \\ &\Longleftrightarrow \boldsymbol{A} \cdot \mathsf{enc}(j_{i+1}, \boldsymbol{v}_{i+1}) + \boldsymbol{A} \cdot \mathsf{enc}(\bar{j}_{i+1}, \boldsymbol{w}_{i+1}) = \boldsymbol{G} \cdot \boldsymbol{v}_i \bmod q, \end{aligned} \tag{4}$$

where we use $\boldsymbol{G}$ to denote the "powers-of-2" matrix $\boldsymbol{G} = \boldsymbol{I}_n \otimes (1\ 2 \cdots 2^{k-1})$. After making the modifications: employ operators $\mathsf{ext}_2(\cdot)$ and $\mathsf{enc}(\cdot,\cdot)$ to extend the targeted vectors $\{\boldsymbol{v}_i\}_i$ and $\{\boldsymbol{w}_i\}_i$, respectively. Then, extend all the involved public matrices accordingly, which allows equally rewriting equation (4) as one that is appropriate for Stern-like zero-knowledge proof [31]. Finally, the desired proof for this system is achieved in ZK by imposing the permutation $F^{(t)}_{\boldsymbol{b},\pi}(\cdot)$ on the extended secret vectors.

3.2 Integrated stern-type zero-knowledge argument system

In this subsection, we first construct the integrated zero-knowledge argument system named as Π_{GE} that is adaptive to both accumulated values and hidden matrices, then the system will be applied in the construction of our group encryption scheme in Section 4. In order for the argument system to be more general, we take the unifying strategy for equations used in [4,7] to run our system. In particular, we still write the unified equation involved with two distinct primes p and q as $\boldsymbol{M} \cdot \boldsymbol{t} = \boldsymbol{u} \bmod Q$ for simplicity, i.e., the presented general equation implicitly consists of two components $\boldsymbol{M}_1 \cdot \boldsymbol{t}_1 = \boldsymbol{u}_1 \bmod p$ and $\boldsymbol{M}_2 \cdot \boldsymbol{t}_2 = \boldsymbol{u}_2 \bmod q$ unless otherwise specified. Based on the specifications, we consider the following system with 10 modular equations:

$$\begin{cases} \boldsymbol{u}_1 = \boldsymbol{M}_{1,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{1,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{1,13} \cdot \boldsymbol{t}_{13} \bmod p, \\ \quad \vdots \\ \boldsymbol{u}_6 = \boldsymbol{M}_{6,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{6,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{6,13} \cdot \boldsymbol{t}_{13} \bmod p, \\ \boldsymbol{u}_7 = \boldsymbol{M}_{7,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{7,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{7,13} \cdot \boldsymbol{t}_{13} \bmod q, \\ \quad \vdots \\ \boldsymbol{u}_{10} = \boldsymbol{M}_{10,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{10,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{10,13} \cdot \boldsymbol{t}_{13} \bmod q. \end{cases} \tag{5}$$

In the above, the matrices and vectors shown as $\{\boldsymbol{M}_{i,j}\}_{(i,j)\in[10]\times[13]}$, $\{\boldsymbol{u}_i\}_{i\in[10]}$ are publicly known, and some of their choices are possibly zero. Our task is to provide the proof for secret vectors $\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{13}$ in ZK, which satisfies (5) while obeying the following constraints:

(1) $\boldsymbol{t}_1\in\{0,1\}^{n\bar{m}k_2}$, $\boldsymbol{t}_2\in\{0,1\}^{nk_2}$, and $\mathsf{expand}^{\otimes}(\boldsymbol{t}_1,\boldsymbol{t}_2)\in\{0,1\}^{4n\bar{m}k_2^2}$. (These vectors are generated via the use of technologies shown in Subsection 3.1).

(2) Vectors $\boldsymbol{t}_4,\ldots,\boldsymbol{t}_{10}$ are binary, where $\boldsymbol{t}_4$ and $\boldsymbol{t}_5$ respectively have complicated representations like $(\mathsf{enc}^{\mathrm{T}}(j_1,\boldsymbol{v}_1),\mathsf{enc}^{\mathrm{T}}(\bar{j}_1,\boldsymbol{w}_1),\ldots,\mathsf{enc}^{\mathrm{T}}(j_\ell,\boldsymbol{v}_\ell),\mathsf{enc}^{\mathrm{T}}(\bar{j}_\ell,\boldsymbol{w}_\ell))^{\mathrm{T}}$ and $(\boldsymbol{u}^{\mathrm{T}},{\boldsymbol{v}_1}^{\mathrm{T}},\ldots,\boldsymbol{v}_{\ell-1}^{\mathrm{T}})^{\mathrm{T}}$ constructed in equation system (15).

(3) Vectors $\boldsymbol{t}_{11}$, $\boldsymbol{t}_{12}$, $\boldsymbol{t}_{13}$ are provided in infinity form.

Towards achieving our goal, we perform the following operations.

– Firstly, employ the decompositions shown in Subsection 3.1 to decompose all vectors that possess norm bigger than 1 into norm-1 ones.

– Secondly, extend the above fresh norm-1 vectors into those invariants under random permutations given in Subsection 3.1. As a necessary step, the public matrices $\{\boldsymbol{M}_{i,j}\}$ are accordingly changed to maintain these equations.

– Thirdly, build $\boldsymbol{M}\cdot\boldsymbol{t}=\boldsymbol{u}\bmod Q$ that unifies all the present equations, where $\boldsymbol{t}$ is obtained via the concatenation of newly generated witness-vectors and Q is adaptively taken as p and q according to the above system (5).

– Finally, utilize a composite permutation and perform a Stern-type protocol, then the proof for the general equation $\boldsymbol{M}\cdot\boldsymbol{t}=\boldsymbol{u}\bmod Q$ is accomplished.

In essence, the above 4-step strategy is an abstraction of the central idea of running Stern-type protocols for relations from lattices [35, 36]: performing appropriate preprocess to witness-vectors to make the generated vectors have invariant weights under any randomly chosen permutations, rewriting these vectors into a unified form, and finally running Stern-like protocol as usual. The concrete steps are seen below.

The first step imposes function $\mathsf{vdec}'(\cdot)$ shown in Subsection 3.1 on vectors $\boldsymbol{t}_{11}$, $\boldsymbol{t}_{12}$, and $\boldsymbol{t}_{13}$, then vector $\boldsymbol{t}_i$ with dimension $\mathfrak{m}_i$ and infinity norm bound β_i is decomposed into $\boldsymbol{t}_i'=\mathsf{vdec}'_{\mathfrak{m}_i,\beta_i}\in\{-1,0,1\}^{\mathfrak{m}_i\delta_{\beta_i}}$. This holds that $\boldsymbol{H}_{\mathfrak{m}_i,\beta_i}\cdot\boldsymbol{t}_i'=\boldsymbol{t}_i$.

The second step executes the following encoding and extending operations.

– Encode $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, and $\boldsymbol{t}_8$: First, compute $\boldsymbol{t}_1''=\mathsf{enc}_{n\bar{m}k_2}(\boldsymbol{t}_1)$ and $\boldsymbol{t}_2''=\mathsf{enc}_{nk_2}(\boldsymbol{t}_2)$ followed by $\boldsymbol{t}_3''=\mathsf{expand}^{\otimes}(\boldsymbol{t}_1,\boldsymbol{t}_2)$, then, compute $\boldsymbol{t}_8''=\mathsf{enc}_\ell(\boldsymbol{t}_8)$, whose knowledge is preserved by the "one-time pad" permuting techniques in (1) and (3).

– Extend $\{0,1\}$-vectors $\boldsymbol{t}_4,\ldots,\boldsymbol{t}_{10}$ (excluding $\boldsymbol{t}_8$) and $\boldsymbol{t}_{11}',\boldsymbol{t}_{12}',\boldsymbol{t}_{13}'$. For each $i\in[6,10]$ but $i=8$, extend the vector $\boldsymbol{t}_i$ to $\boldsymbol{t}_i''=\mathsf{ext}_2(\boldsymbol{t}_i)\in\mathsf{B}_{2\mathfrak{m}_i}^{\mathfrak{m}_i}$ assuming dimension $\mathfrak{m}_i$. Similarly, vector $\boldsymbol{t}_i''=\mathsf{ext}_3(\boldsymbol{t}_i')\in\mathsf{B}_{3\mathfrak{m}_i\delta_{\beta_i}}^{\mathfrak{m}_i\delta_{\beta_i}}$ with dimension $\mathfrak{m}_i\delta_{\beta_i}$ is obtained for each $i\in[11,13]$. By permutations shown in Subsection 3.1, proving the knowledge of vectors $\{\boldsymbol{t}_i''\}_{i=6}^{13}$ (except for $i=8$) in ZK is achieved. But for $i=4,5$, we need a much more sophisticated treatment: set $\boldsymbol{t}_6=\boldsymbol{v}_\ell$, and extend $\{\boldsymbol{v}_j\}_{j=1}^{\ell}$, $\{\boldsymbol{w}_j\}_{j=1}^{\ell}$ via $\mathsf{ext}_2(\cdot)$ to $\{\boldsymbol{v}_j''\}_{j=1}^{\ell},\{\boldsymbol{w}_j''\}_{j=1}^{\ell}\in\mathsf{B}_{m_1}^{nk_1}$, respectively, then build $\boldsymbol{t}_4''=(\mathsf{enc}^{\mathrm{T}}(j_1,\boldsymbol{v}_1''),\mathsf{enc}^{\mathrm{T}}(\bar{j}_1,\boldsymbol{w}_1''),\ldots,\mathsf{enc}^{\mathrm{T}}(j_\ell,\boldsymbol{v}_\ell''),\mathsf{enc}^{\mathrm{T}}(\bar{j}_\ell,\boldsymbol{w}_\ell''))^{\mathrm{T}}$ and $\boldsymbol{t}_5''=(\boldsymbol{u}^{\mathrm{T}},{\boldsymbol{v}_1''}^{\mathrm{T}},\ldots,\boldsymbol{v}_{\ell-1}''^{\mathrm{T}})^{\mathrm{T}}$ by these newly computed vectors.

To construct permutations suitable to vectors $\boldsymbol{t}_4''$ and $\boldsymbol{t}_5''$, we use ComMix_1 to represent the set of all size-$2m_1\ell$ bit vectors that meanwhile possess the form $(\mathsf{enc}^{\mathrm{T}}(j_1,\boldsymbol{v}_1^*),\mathsf{enc}^{\mathrm{T}}(\bar{j}_1,\boldsymbol{w}_1^*),\ldots,\mathsf{enc}^{\mathrm{T}}(j_\ell,\boldsymbol{v}_\ell^*),\mathsf{enc}^{\mathrm{T}}(\bar{j}_\ell,\boldsymbol{w}_\ell^*))^{\mathrm{T}}$, and similarly set ComMix_2 as that of vectors in $\{0,1\}^{(2\ell-1)nk_1}$ with the form $(\boldsymbol{u}^{\mathrm{T}},{\boldsymbol{v}_1^*}^{\mathrm{T}},\ldots,\boldsymbol{v}_{\ell-1}^{*\mathrm{T}})^{\mathrm{T}}$ for $\{\boldsymbol{v}_i^*\}_{i=1}^{\ell},\{\boldsymbol{w}_i^*\}_{i=1}^{\ell}\in\mathsf{B}_{m_1}^{nk_1}$. It is trivial to check that vectors $\boldsymbol{t}_4''\in\mathsf{ComMix}_1$ and $\boldsymbol{t}_5''\in\mathsf{ComMix}_2$, respectively. Now, for $i\in[1,\ell]$, pick random $b_i\in\{0,1\}$; $\pi_i,\phi_i\in\mathcal{S}_{m_1}$ respect to vectors $\{\boldsymbol{v}_i\}_i$ and $\{\boldsymbol{w}_i\}_i$, respectively, and set $\boldsymbol{b}=(b_1,b_1,\ldots,b_\ell,b_\ell)^{\mathrm{T}}$, $\pi^{(1)}=(\pi_1,\phi_1,\ldots,\pi_\ell,\phi_\ell)$, then former is closed under the permutation $F_{\boldsymbol{b},\pi^{(1)}}^{(2\ell)}(\boldsymbol{t}_4'')$ shown as in Subsection 3.1. For the latter, we define a composed permutation $\pi^{(2)}:=(\varepsilon,\pi_1,\ldots,\pi_{\ell-1})$ where the symbol ε is used to represent the identity permutation, then the targeted vector is also closed under the permutation $F_{\pi^{(2)}}^{(\ell)}(\boldsymbol{t}_5'')$ given in Subsection 3.1. All of this gives that

$$\boldsymbol{t}_4''\in\mathsf{ComMix}_1\Longleftrightarrow F_{\boldsymbol{b},\pi^{(1)}}^{(2\ell)}(\boldsymbol{t}_4''),\quad \boldsymbol{t}_5''\in\mathsf{ComMix}_2\Longleftrightarrow F_{\pi^{(2)}}^{(\ell)}(\boldsymbol{t}_5''). \tag{6}$$

– Like the above discussion, dimensions of the witness-vectors are changed, which makes it necessary to perform some modifications on the public matrices $\{\boldsymbol{M}_{i,j}\}$ to maintain the original equations (5). This can be readily achieved by imposing simple operations on primitive public matrices.

By the above statements, we are provided with the following system which is actually equal to the system (5):

$$\begin{cases} \boldsymbol{u}_1 = \boldsymbol{M}''_{1,1}\cdot\boldsymbol{t}''_1 + \boldsymbol{M}''_{1,2}\cdot\boldsymbol{t}''_2 + \cdots + \boldsymbol{M}''_{1,13}\cdot\boldsymbol{t}''_{13} \bmod p, \\ \quad\vdots \\ \boldsymbol{u}_6 = \boldsymbol{M}''_{6,1}\cdot\boldsymbol{t}''_1 + \boldsymbol{M}''_{6,2}\cdot\boldsymbol{t}''_2 + \cdots + \boldsymbol{M}''_{6,13}\cdot\boldsymbol{t}''_{13} \bmod p, \\ \boldsymbol{u}_7 = \boldsymbol{M}''_{7,1}\cdot\boldsymbol{t}''_1 + \boldsymbol{M}''_{7,2}\cdot\boldsymbol{t}''_2 + \cdots + \boldsymbol{M}''_{7,13}\cdot\boldsymbol{t}''_{13} \bmod q. \\ \quad\vdots \\ \boldsymbol{u}_{10} = \boldsymbol{M}''_{10,1}\cdot\boldsymbol{t}''_1 + \boldsymbol{M}''_{10,2}\cdot\boldsymbol{t}''_2 + \cdots + \boldsymbol{M}''_{10,13}\cdot\boldsymbol{t}''_{13} \bmod q. \end{cases} \tag{7}$$

The final step aims to prove secret witness-vectors in ZK and only involves simple algebra knowledge. Let

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{M}''_{1,1} & \boldsymbol{M}''_{1,2} & \dots & \boldsymbol{M}''_{1,13} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{M}''_{6,1} & \boldsymbol{M}''_{6,2} & \dots & \boldsymbol{M}''_{6,13} \\ \boldsymbol{M}''_{7,1} & \boldsymbol{M}''_{7,2} & \dots & \boldsymbol{M}''_{7,13} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{M}''_{10,1} & \boldsymbol{M}''_{10,2} & \dots & \boldsymbol{M}''_{10,13} \end{pmatrix}, \ \boldsymbol{t} = \begin{pmatrix} \boldsymbol{t}''_1 \\ \boldsymbol{t}''_2 \\ \vdots \\ \boldsymbol{t}''_7 \\ \vdots \\ \boldsymbol{t}''_{13} \end{pmatrix}, \ \boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_6 \\ \boldsymbol{u}_7 \\ \vdots \\ \boldsymbol{u}_{10} \end{pmatrix},$$

which directly results into the unified system $\boldsymbol{M}\cdot\boldsymbol{t} = \boldsymbol{u} \bmod Q$.

After making the preparations above, we perform the final step to put forth our protocol. For the targeted vector $\boldsymbol{t}$, we first denote its dimension by D quantified as $D = 2n\bar{m}k_2 + 2nk_2 + 4n\bar{m}k_2^2 + 10nk_1\ell + 2m_2 + 2\ell + 4\widetilde{m} + 6m_2\delta_B + 3\bar{m}\delta_{\beta m_2 B}$, then we define the set of all dimension-D vectors that have the form $\boldsymbol{z} = (\boldsymbol{z}_1^{\mathrm{T}}\|\cdots\|\boldsymbol{z}_{13}^{\mathrm{T}})^{\mathrm{T}}$ consisting of $\{-1,0,1\}$ entries as VALID, where

(1) $\boldsymbol{z}_1 = \mathsf{enc}_{n\bar{m}k_2}(\boldsymbol{y}_1)$, $\boldsymbol{z}_2 = \mathsf{enc}_{nk_2}(\boldsymbol{y}_2)$, and $\boldsymbol{z}_3 = \mathsf{expand}^{\otimes}(\boldsymbol{y}_1,\boldsymbol{y}_2)$ and $\boldsymbol{z}_8 = \mathsf{enc}_{\ell}(\boldsymbol{y}_8)$, for $(\boldsymbol{y}_1,\boldsymbol{y}_2,\boldsymbol{y}_8) \in \{0,1\}^{n\bar{m}k_2}\times\{0,1\}^{nk_2}\times\{0,1\}^{\ell}$.

(2) $\boldsymbol{z}_i \in \mathsf{B}^{\mathtt{m}_i}_{2\mathtt{m}_i}$ for $i\in[6,10]$ (but $i=8$); while for $i\in[11,13]$, $\boldsymbol{z}_i \in \mathsf{B}^{\mathtt{m}_i\delta_{\beta_i}}_{3\mathtt{m}_i\delta_{\beta_i}}$.

(3) Vectors $\boldsymbol{z}_4 \in \mathsf{ComMix}_1$ and $\boldsymbol{z}_5 \in \mathsf{ComMix}_2$.

It is clear that the unified vector $\boldsymbol{t}$ is one of the possible elements of the fresh tailored set VALID. By the above construction, the task that one performs a proof to convince verifiers that he knows vectors $\boldsymbol{t}_1,\dots,\boldsymbol{t}_{13}$ satisfying these specific constraints and system (5) is equal to that demonstrating his possession of vector $\boldsymbol{t}\in\mathsf{VALID}$ satisfies the unified system $\boldsymbol{M}\cdot\boldsymbol{t} = \boldsymbol{u} \bmod Q$. To achieve this goal in ZK, we will execute a Stern-type ZK protocol in which we employ masking techniques shown as uniformly random chosen vectors as well as permutations to hide the targeted $\boldsymbol{t}$. This makes it convenient to define the permutations that are applicable to $\boldsymbol{t}$. Let

$$\begin{aligned} \mathcal{S} = \{0,1\}^{n\bar{m}k_2}\times\{0,1\}^{nk_2}\times\overbrace{\mathcal{S}^{\boldsymbol{v}}_{2m_1}\times\cdots\times\mathcal{S}^{\boldsymbol{v}}_{2m_1}}^{\ell\text{ times}}\times\overbrace{\mathcal{S}^{\boldsymbol{w}}_{2m_1}\times\cdots\times\mathcal{S}^{\boldsymbol{w}}_{2m_1}}^{\ell\text{ times}} \\ \times\mathcal{S}_{2\mathtt{m}_6}\times\mathcal{S}_{2\mathtt{m}_7}\times\{0,1\}^{\ell}\times\mathcal{S}_{2\mathtt{m}_9}\times\mathcal{S}_{2\mathtt{m}_{10}}\times\mathcal{S}_{3\mathtt{m}_{11}\delta_{\beta_{11}}}\times\mathcal{S}_{3\mathtt{m}_{12}\delta_{\beta_{12}}}\times\mathcal{S}_{3\mathtt{m}_{13}\delta_{\beta_{13}}}. \end{aligned} \tag{8}$$

Set $\pi^{(1)} = \{\pi^{\boldsymbol{v}}_i,\phi^{\boldsymbol{w}}_i\}_{i=1}^{\ell}$, $\pi^{(2)} = \{\varepsilon,\{\pi^{\boldsymbol{v}}_i\}_{i=1}^{\ell-1}\}$, $\pi_6 = \pi^{\boldsymbol{v}}_{\ell}$, $\boldsymbol{b}_8 = (b_1^{(8)},\dots,b_{\ell}^{(8)})^{\mathrm{T}}$, $\boldsymbol{b}'_8 = (b_1^{(8)},b_1^{(8)},\dots,b_{\ell}^{(8)},b_{\ell}^{(8)})^{\mathrm{T}}$. We associate each element $\pi = (\boldsymbol{b}_1,\boldsymbol{b}_2,\pi^{(1)},\pi^{(2)},\pi_6,\pi_7,\boldsymbol{b}_8,\boldsymbol{b}'_8,\pi_9,\dots,\pi_{13})$ with the permutation Γ_π, which transforms vector $\boldsymbol{t} = (\boldsymbol{t}_1^{\mathrm{T}},\dots,\boldsymbol{t}_{13}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{Z}^D$ where each block $\boldsymbol{t}_i$ has the same length as $\boldsymbol{t}''_i$ for all $i\in[13]$ into vector

$$\begin{aligned} \Gamma_\pi(\boldsymbol{t}) = \big(&T^{(n\bar{m}k_2)}_{\boldsymbol{b}_1}(\boldsymbol{t}_1)\|T^{(nk_2)}_{\boldsymbol{b}_2}(\boldsymbol{t}_2)\|P_{\boldsymbol{b}_1,\boldsymbol{b}_2}(\boldsymbol{t}_3)\|F^{(2\ell)}_{\boldsymbol{b}'_8,\pi^{(1)}}(\boldsymbol{t}_4)\|F^{(\ell)}_{\pi^{(2)}}(\boldsymbol{t}_5) \\ &\|\pi_6(\boldsymbol{t}_6)\|\pi_7(\boldsymbol{t}_7)\|T^{(\ell)}_{\boldsymbol{b}_8}(\boldsymbol{t}_8)\|\pi_9(\boldsymbol{z}_9)\|\cdots\|\pi_{13}(\boldsymbol{t}_{13})\big). \end{aligned}$$

It is trivial that, by the permutations shown in Subsection 3.1, the following relation holds for any sampled $\pi\in\mathcal{S}$.

$$\boldsymbol{t}\in\mathsf{VALID} \Longleftrightarrow \Gamma_\pi(\boldsymbol{t})\in\mathsf{VALID}.$$

1. **Commitment.** $\mathcal{P}$ samples $\boldsymbol{r}_t \hookleftarrow U(\mathbb{Z}_Q^D)$, $\pi \hookleftarrow U(\mathcal{S})$ and randomness ρ_1, ρ_2, ρ_3 for COM, then sends CMT$= (C_1, C_2, C_3)$ to the verifier, where

$$C_1 = \mathsf{COM}(\pi, \boldsymbol{M} \cdot \boldsymbol{r}_t \,; \rho_1), C_2 = \mathsf{COM}(\Gamma_\pi(\boldsymbol{r}_t); \rho_2), C_3 = \mathsf{COM}(\Gamma_\pi(\boldsymbol{t} + \boldsymbol{r}_t); \rho_3).$$

2. **Challenge.** Receiving CMT, $\mathcal{V}$ sends a challenge $\mathrm{Ch} \hookleftarrow \{1, 2, 3\}$ to $\mathcal{P}$.
3. **Response.** Depending on Ch, $\mathcal{P}$ sends response RSP computed as follows:
   - $\mathrm{Ch} = 1$: Let $\boldsymbol{z}_t = \Gamma_\pi(\boldsymbol{t})$, $\boldsymbol{z}_r = \Gamma_\pi(\boldsymbol{r}_t)$, and RSP$= (\boldsymbol{z}_t, \boldsymbol{z}_r, \rho_2, \rho_3)$.
   - $\mathrm{Ch} = 2$: Let $\pi_2 = \pi$, $\boldsymbol{y}_2 = \boldsymbol{t} + \boldsymbol{r}_t$, and RSP$= (\pi_2, \boldsymbol{y}_2, \rho_1, \rho_3)$.
   - $\mathrm{Ch} = 3$: Let $\pi_3 = \pi$, $\boldsymbol{y}_3 = \boldsymbol{r}_t$, and RSP$= (\pi_3, \boldsymbol{y}_3, \rho_1, \rho_2)$.

Verification. Receiving RSP, the verifier proceeds as follows:

- $\mathrm{Ch} = 1$: Check that $\boldsymbol{z}_t \in \mathsf{VALID}$ and $C_2 = \mathsf{COM}(\boldsymbol{z}_r \,; \rho_2), C_3 = \mathsf{COM}(\boldsymbol{z}_t + \boldsymbol{z}_r \,; \rho_3)$.
- $\mathrm{Ch} = 2$: Check that $C_1 = \mathsf{COM}(\pi_2, \boldsymbol{M} \cdot \boldsymbol{y}_2 - \boldsymbol{u}; \rho_1), C_3 = \mathsf{COM}(\Gamma_{\pi_2}(\boldsymbol{y}_2); \rho_3)$.
- $\mathrm{Ch} = 3$: Check that $C_1 = \mathsf{COM}(\pi_3, \boldsymbol{M} \cdot \boldsymbol{y}_3; \rho_1), C_2 = \mathsf{COM}(\Gamma_{\pi_3}(\boldsymbol{y}_3); \rho_2)$.

In each case, the verifier outputs 1 if and only if all conditions hold.

Figure 1 Our zero-knowledge argument of knowledge for the GE.

In addition, given any $\boldsymbol{t} \in \mathsf{VALID}$ and $\pi \hookleftarrow \mathcal{S}$, we will find that $\Gamma_\pi(\boldsymbol{t}) \in \mathsf{VALID}$, which is sufficient to provide a ZK proof for $\boldsymbol{t} \in \mathsf{VALID}$ under the Stern's protocol. Furthermore, proving the secret vector $\boldsymbol{t}$ satisfying $\boldsymbol{M} \cdot \boldsymbol{t} = \boldsymbol{u} \bmod Q$ will be accomplished by showing that $\boldsymbol{M} \cdot (\boldsymbol{t} + \boldsymbol{r_t}) - \boldsymbol{u} = \boldsymbol{M} \cdot \boldsymbol{r_t} \bmod Q$ to the verifier.

For clarity, we depict the interaction executed between prover $\mathcal{P}$ and verifier $\mathcal{V}$ in Figure 1 by specifying their respective executions. Prior to the interaction, both parties will get the public matrix $\boldsymbol{M}$ and vector $\boldsymbol{u}$ from the public inputs. Here, $\mathcal{P}$ constructs the desired vector $\boldsymbol{t}$ using the secret inputs, as described above. The protocol plays the central role under the string commitment scheme COM proposed in [37] that has a few appealing properties.

For completeness, we summarize our protocol's properties in Theorem 1, whose concrete statements are found in the subsequent discussion and lemmas. Note that the proof of Stern-like ZKAoK for a unified system is trivial, similar to that of [4, 7] by adopting some techniques from [13]. Thus, due to space limitations, we do not address in detail.

Theorem 1. Given a string commitment scheme COM that both has the statistically hiding and computationally binding properties, then the interactive protocol provided above is a statistical ZKAoK that shares perfect completeness, soundness error 2/3, and communication cost $\widetilde{\mathcal{O}}(D \log Q)$.

Completeness and communication cost. By the previous sections, it is easily checked that the present protocol is complete, and $\mathcal{V}$ outputs 1 if $\mathcal{P}$ honestly follows the protocol. In addition, it is also observed that the communication cost is $\widetilde{\mathcal{O}}(D \log Q)$ bits.

Lemma 1 (Zero-knowledgeness)**.** Suppose that the COM has the statistically hiding property, then there is an efficient simulator that takes the $(\boldsymbol{M}, \boldsymbol{u})$ as input, and produces an accepted transcript that keeps the negligible statistical distance to that generated by the real prover.

Lemma 2 (Argument of knowledge)**.** Assume the COM having the computationally binding property, given the commitment CMT, then there is a polynomial-time knowledge extractor $\mathcal{K}$ that gets 3 valid responses $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ to the corresponding challenge values $\mathrm{Ch} = 1, 2, 3$ as input, respectively, and generates a $\boldsymbol{t}' \in \mathsf{VALID}$ satisfying the equation $\boldsymbol{M} \cdot \boldsymbol{t}' = \boldsymbol{u} \bmod Q$.

4 Our proposed group encryptions from lattices

In this section, we show how to apply the LLNW Merkle-tree accumulators [13, 14] and the Stern-type argument system described in Subsection 3.2 to construct our group encryption scheme over lattice assumptions. In terms of efficiency, the proposed scheme is comparable to the previously proposed scheme [4] since only one lattice trapdoor is required. Indeed, the presented scheme does not use the signature technique any longer to identify the group membership, and it hides the targeted group users' identities via the multi-bit version of the Regev encryption scheme, all of which eliminate the need to use lattice trapdoors.

4.1 Description of the scheme

As in [4], our GE scheme also allows encrypting witness for the Inhomogeneous SIS relation R_{ISIS} given by $((\boldsymbol{A}_R, \boldsymbol{u}_R), \boldsymbol{m}) \in (\mathbb{Z}_q^{n\times m_2} \times \mathbb{Z}_q^n) \times \{0,1\}^{m_2}$ with $\boldsymbol{u}_R = \boldsymbol{A}_R \cdot \boldsymbol{m} \bmod q$, which is a lattice version in the same spirit as that of [1]. Given the maximum number of prospective members $N = 2^\ell = \mathsf{poly}(\lambda)$, we make steps to build the GE scheme.

• $\mathsf{SETUP}_{\mathrm{init}}$ (1^λ): This procedure conducts the following.

(1) Select integer $n = \mathcal{O}(\lambda)$ and primes $p = \widetilde{\mathcal{O}}(n^{1.5})$ and $q = \widetilde{\mathcal{O}}(n^4)$ followed by $k_1 = \lceil \log p \rceil$, $k_2 = \lceil \log q \rceil$, respectively. Then, for each $i \in \{1,2\}$, set $m_i = 2nk_i$, $\widetilde{m} = 2(n+\ell)k_1$ and $m_2 = 2\bar{m}$, and build a discrete distribution χ bounded by $B = \sqrt{n}\omega(\log n)$ over the integer ring $\mathbb{Z}$.

(2) Select positive $\sigma = \Omega(\sqrt{n \log q} \log n)$ to build a discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ of which samples are bounded by $\beta = \sigma \cdot \omega(\log n)$.

(3) Choose the public parameters $\mathsf{par}_{\mathsf{COM}}$ for the string commitment scheme [37] that serves as the construction of the ZKAoK system used in $\langle \mathcal{P}, \mathcal{V} \rangle$, and take $\kappa = \omega(\log \lambda)$ as the repetition times of protocol.

(4) Sample a one-time signature scheme $\mathcal{OTS} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with strong unforgeability whose verification key lives in $\mathbb{Z}_q^n$.

(5) Take FRD: $\mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ as the full-rank difference operator [38].

(6) Choose a matrix $\boldsymbol{F} \hookleftarrow U(\mathbb{Z}_p^{n\times n\bar{m}k_2})$ which hashes the public keys of group users from $\mathbb{Z}_q^{n\times\bar{m}}$ to $\mathbb{Z}_p^n$.

(7) Set basic matrices $\boldsymbol{G}_i = \boldsymbol{I}_n \otimes [1\ 2 \cdots 2^{k_i-1}]$ for each $i \in \{1,2\}$. Choose a matrix $\boldsymbol{A} = [\boldsymbol{A}_0 | \boldsymbol{A}_1] \hookleftarrow U(\mathbb{Z}_p^{n\times m_1})$ consisting of two same-size blocks that will be used in the construction of accumulator, and select matrices $\bar{\boldsymbol{A}}$, $\boldsymbol{U} \hookleftarrow U(\mathbb{Z}_q^{n\times m_2})$ used in encrypting messages.

Output $\mathsf{pp} = \{\lambda, n, q, k_1, k_2, m_1, m_2, B, \chi, \sigma, \beta, N, \kappa, \mathcal{OTS}, \mathsf{par}_{\mathsf{COM}}, \mathsf{FRD}, \boldsymbol{A}, \bar{\boldsymbol{A}}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{F}, \boldsymbol{U}\}$.

• $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{pp})$: Sample a one-time signature with verification/signing key pair $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \in (\mathbb{Z}_p^n \times \{0,1\}^{m_1})^{m_1}$ for the GM, where $\mathsf{sk}_{\mathsf{GM}}$ consists of m_1 binary vectors $\mathsf{sk}_{\mathsf{GM}}^{(i)} \in \{0,1\}^{m_1}$, each of which matches that of $\mathsf{pk}_{\mathsf{GM}}$ over $\mathbb{Z}_p^n$ under the mapping $\mathsf{pk}_{\mathsf{GM}}^{(i)} = \boldsymbol{A} \cdot \mathsf{sk}_{\mathsf{GM}}^{(i)}$.

• $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{pp})$: This procedure picks a matrix $\boldsymbol{B} \hookleftarrow U(\mathbb{Z}_p^{n\times\widetilde{m}})$, also does $\boldsymbol{S}_i \hookleftarrow U(\mathbb{Z}_p^{n\times\ell})$ and $\boldsymbol{E}_i \leftarrow \chi^{\ell\times\widetilde{m}}$ for $i \in \{1,2\}$, and computes $\boldsymbol{P}_i = \boldsymbol{S}_i^{\mathrm{T}} \cdot \boldsymbol{B} + \boldsymbol{E}_i \in \mathbb{Z}_p^{\ell\times\widetilde{m}}$, resulting a key pair $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) = ((\boldsymbol{B}, \boldsymbol{P}_1, \boldsymbol{P}_2), \boldsymbol{S}_1)$ for the OA.

When GM receives $\mathsf{pk}_{\mathsf{OA}}$ sent from the OA, it executes the following:

(1) Build table $\mathbf{reg}$: $= (\mathbf{reg}[0], \ldots, \mathbf{reg}[N-1])$ initialized as $\mathbf{reg}[i] = \mathbf{0}^{nk}$ for each $i \in [0, N-1]$. It will write the records of the registered public keys.

(2) Build a Merkle tree $\mathcal{T}$ based on the records of the table $\mathbf{reg}$. It is noted that all $\mathbf{reg}[i]$'s are zero at the outset and are changed with users' public keys by the GM when one successfully joins the group.

(3) Set the counter as $c := 0$.

Later on, the GM builds and publishes the group public key gpk=(pp,$\mathsf{pk}_{\mathsf{GM}}$,$\mathsf{pk}_{\mathsf{OA}}$), while $\mathcal{T}$ as well as c is kept by himself.

• $\mathsf{UKGEN}(\mathsf{pp})$: For each $j \in [0, N-1]$, user U_j samples $\boldsymbol{T}_j \leftarrow D_{\mathbb{Z}^{\bar{m}_2},\sigma}^{\bar{m}}$ and computes a statistically uniform matrix $\boldsymbol{B}_j = \bar{\boldsymbol{A}} \cdot \boldsymbol{T}_j \in \mathbb{Z}_q^{n\times\bar{m}}$, resulting into a key pair $(\mathsf{pk}_j, \mathsf{sk}_j) = (\boldsymbol{B}_j, \boldsymbol{T}_j)$ with an associated binary hash value $\boldsymbol{p}_j = \mathsf{bin}(\boldsymbol{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\boldsymbol{B}_j)) \in \{0,1\}^{nk_1}$. We say that w.l.o.g. all honestly generated pk_j's are non-zero and pairwise distinct, since the probability that one takes $\boldsymbol{T}_j = \mathbf{0}$, or $\boldsymbol{T}_j = \boldsymbol{T}_{j'}$ for some $j \neq j'$, or finds a solution to the average-case lattice problem $\mathsf{SIS}_{n,m_1,p,1}$ underlying the Merkle tree is negligible.

• $\langle \mathsf{JOIN}(\mathsf{gpk}, \mathsf{pk}_j, \mathsf{sk}_j); \mathsf{ISSUE}(\mathsf{sk}_{\mathsf{GM}}, \mathsf{pk}_j) \rangle$: When one having a key pair $(\mathsf{pk}_j, \mathsf{sk}_j)$ with the hash value $\boldsymbol{p}_j$ wants to become a valid group user, he transmits $\boldsymbol{p}_j$ to the GM who proceeds the following procedures with him after the request is accepted:

(1) GM sets a member identifier $\mathsf{uid} = \mathsf{bin}(c) = \mathsf{bin}(j) \in \{0,1\}^\ell$ for the user, and executes the following:

– Write the valid hash value into the table $\mathbf{reg}$ as $\mathbf{reg}[c] := \boldsymbol{p}_j$.

– Increase the counter $c := c + 1$ till the maximum expected $N - 1$.

(2) When $c = N-1$, the GM runs the algorithm $\mathsf{TAcc}_{\boldsymbol{A}}(R)$ to build $\mathcal{T}$ based on $R = (\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{N-1})$ and obtains the root value $\boldsymbol{u} \in \{0,1\}^{nk_1}$. Then, for each $j \in [0, N-1]$, invoke the procedure $\mathsf{TWitness}_{\boldsymbol{A}}(R, \boldsymbol{p}_j)$ to generate the corresponding witness

$$w^{(j)} = \big((j_1, \ldots, j_\ell), (\boldsymbol{w}_\ell^{(j)}, \ldots, \boldsymbol{w}_1^{(j)}) \in \{0,1\}^\ell \times (\{0,1\}^{nk_1})^\ell\big)$$

that demonstrates $\boldsymbol{p}_j$ is actually accumulated in $\boldsymbol{u}$. Finally, issue the signature $\sigma_{\boldsymbol{u}}$ on root $\boldsymbol{u}$ under $\mathsf{sk}_{\mathsf{GM}}$, as well as $\boldsymbol{u}$ and witnesses $\{w^{(j)}\}_{j=1}^{N}$.

(3) User checks the validity of $\sigma_{\boldsymbol{u}}$ and $w^{(j)}$ and outputs $\perp$ if they are unaccepted. Otherwise, set $\mathsf{cert}_j = w^{(j)}$ as the certificate of pk_j.

• $\langle \mathcal{G}_r, \mathsf{sample}_{\mathcal{R}} \rangle$: Algorithm $\mathcal{G}_r$ outputs $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}) = (\boldsymbol{A}_R, \varepsilon)$, then the sampler $\mathsf{sample}_{\mathcal{R}}$ takes $\mathsf{pk}_{\mathcal{R}}$ as input to output a tuple $((\boldsymbol{A}_R, \boldsymbol{u}_R), \boldsymbol{m})$ satisfying $\boldsymbol{u}_R = \boldsymbol{A}_R \cdot \boldsymbol{m}$.

• $\mathsf{ENC}(\mathsf{gpk}, \mathsf{pk}_j, \mathsf{cert}_j, \boldsymbol{m}, L)$: To encrypt message $\boldsymbol{m}$ obtained from algorithm $\mathsf{sample}_{\mathcal{R}}$, parse $\mathsf{pk}_{\mathsf{OA}}$ as $(\boldsymbol{B}, \boldsymbol{P}_1, \boldsymbol{P}_2)$ and cert_j as $w^{(j)}$ for some $j \in [0, N-1]$.

(1) Select a key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathsf{Gen}(1^\lambda)$ with $\mathsf{VK} \in \mathbb{Z}_q^n$ for the one-time signature.

(2) Obtain the full-rank-difference result $\boldsymbol{H}_{\mathsf{VK}} = \mathsf{FRD}(\mathsf{VK}) \in \mathbb{Z}_q^{n \times n}$ of the above generated $\mathsf{VK} \in \mathbb{Z}_q^n$.

(3) Encrypt a message $\boldsymbol{m} \in \{0,1\}^{m_2}$ under the user U_j's public key $\mathsf{pk}_j \in \mathbb{Z}_q^{n \times \bar{m}}$ as follows.

(a) Sample $\boldsymbol{s}_{\mathsf{rec}} \hookleftarrow U(\mathbb{Z}_q^n), \boldsymbol{R}_{\mathsf{rec}} \leftarrow D_{\mathbb{Z},\sigma}^{m_2 \times \bar{m}}$, and $\boldsymbol{x}_{\mathsf{rec}}, \boldsymbol{y}_{\mathsf{rec}} \leftarrow \chi^{m_2}$, then compute $\boldsymbol{z}_{\mathsf{rec}} = \boldsymbol{R}_{\mathsf{rec}}^{\mathrm{T}} \cdot \boldsymbol{y}_{\mathsf{rec}} \in \mathbb{Z}^{\bar{m}}$.

(b) Compute

$$\begin{cases} \boldsymbol{c}_{\mathsf{rec}}^{(1)} = \bar{\boldsymbol{A}}^{\mathrm{T}} \cdot \boldsymbol{s}_{\mathsf{rec}} + \boldsymbol{y}_{\mathsf{rec}} \bmod q, \\ \boldsymbol{c}_{\mathsf{rec}}^{(2)} = (\boldsymbol{B}_j + \boldsymbol{H}_{\mathsf{VK}} \cdot \boldsymbol{G}_2)^{\mathrm{T}} \cdot \boldsymbol{s}_{\mathsf{rec}} + \boldsymbol{z}_{\mathsf{rec}} \bmod q, \\ \boldsymbol{c}_{\mathsf{rec}}^{(3)} = \boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{s}_{\mathsf{rec}} + \boldsymbol{x}_{\mathsf{rec}} + \left\lfloor \frac{q}{2} \right\rceil \cdot \boldsymbol{m} \bmod q, \end{cases} \tag{9}$$

then set $\boldsymbol{c}_{\mathsf{rec}} = (\boldsymbol{c}_{\mathsf{rec}}^{(1)}, \boldsymbol{c}_{\mathsf{rec}}^{(2)}, \boldsymbol{c}_{\mathsf{rec}}^{(3)}) \in \mathbb{Z}_q^{m_2} \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^{m_2}$, which results an ABB ciphertext [38] associated with the given tag VK.

(4) Use the dual Regev encryption mechanism to encrypt $\boldsymbol{j} = (j_1, \ldots, j_\ell)^{\mathrm{T}} \in \{0,1\}^\ell$. For each $i \in \{1,2\}$, sample $\boldsymbol{r}_{\mathsf{oa}}^{(i)} \hookleftarrow \{0,1\}^{\bar{m}}$ and compute $\boldsymbol{c}_{\mathsf{oa}}^{(i)} = (\boldsymbol{c}_{\mathsf{oa}}^{(i,1)}, \boldsymbol{c}_{\mathsf{oa}}^{(i,2)})$ as follows:

$$\begin{aligned} \boldsymbol{c}_{\mathsf{oa}}^{(i,1)} &= \boldsymbol{B} \cdot \boldsymbol{r}_{\mathsf{oa}}^{(i)} \bmod p, \\ \boldsymbol{c}_{\mathsf{oa}}^{(i,2)} &= \boldsymbol{P}_i \cdot \boldsymbol{r}_{\mathsf{oa}}^{(i)} + \left\lfloor \frac{p}{2} \right\rceil \cdot \boldsymbol{j} \bmod p, \end{aligned} \tag{10}$$

which leads $\boldsymbol{c}_{\mathsf{oa}} = (\boldsymbol{c}_{\mathsf{oa}}^{(1)}, \boldsymbol{c}_{\mathsf{oa}}^{(2)}) \in (\mathbb{Z}_p^n \times \mathbb{Z}_p^\ell)^2$.

(5) Compute a one-time signature $\Sigma = \mathsf{Sig}(\mathsf{SK}, (\mathsf{c}_{\mathsf{rec}}, \mathsf{c}_{\mathsf{oa}}, L))$.

Return the final ciphertext as

$$\Psi = \big(\mathsf{VK}, \boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, \Sigma\big), \tag{11}$$

and the secret state information $\mathsf{coins}_\Psi = (\boldsymbol{s}_{\mathsf{rec}}, \boldsymbol{R}_{\mathsf{rec}}, \boldsymbol{x}_{\mathsf{rec}}, \boldsymbol{y}_{\mathsf{rec}}, \boldsymbol{r}_{\mathsf{oa}}^{(1)}, \boldsymbol{r}_{\mathsf{oa}}^{(2)})$.

• $\mathsf{DEC}(\mathsf{sk}_j, \Psi, L)$: The decryptor takes the following executions:

(1) Return $\perp$ if $\mathsf{Ver}(\mathsf{VK}, \Sigma, (\boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, L)) = 0$. Otherwise, parse the secret key sk_j as $\boldsymbol{T}_j \in \mathbb{Z}^{m_2 \times \bar{m}}$ and the ciphertext Ψ as in (11), meanwhile define a fresh matrix $\boldsymbol{B}_{\mathsf{VK}} = \boldsymbol{B}_j + \mathsf{FRD}(\mathsf{VK}) \cdot \boldsymbol{G}_2 \in \mathbb{Z}_q^{n \times \bar{m}}$.

(2) Decrypt $\boldsymbol{c}_{\mathsf{rec}}$ with the sampled key below associated with the specific VK:

(a) Set $\boldsymbol{B}_{j,\mathsf{VK}} = [\bar{\boldsymbol{A}} | \boldsymbol{B}_{\mathsf{VK}}] = [\bar{\boldsymbol{A}} | \bar{\boldsymbol{A}} \cdot \boldsymbol{T}_j + \mathsf{FRD}(\mathsf{VK}) \cdot \boldsymbol{G}_2] \in \mathbb{Z}_q^{n \times (m_2 + \bar{m})}$. Utilize $\boldsymbol{T}_j$ and the publicly known trapdoor $\boldsymbol{T}_{\boldsymbol{G}_2}$ of $\boldsymbol{G}_2$, then invoke the $\mathsf{SampleRight}$ algorithm in [38] to sample a small norm matrix $\boldsymbol{E}_{\mathsf{VK}} \in \mathbb{Z}_q^{(m_2+\bar{m}) \times m_2}$ that satisfies $\boldsymbol{B}_{j,\mathsf{VK}} \cdot \boldsymbol{E}_{\mathsf{VK}} = \boldsymbol{U} \bmod q$.

(b) Compute

$$\boldsymbol{m} = \left\lfloor \left(\boldsymbol{c}_{\mathsf{rec}}^{(3)} - \boldsymbol{E}_{\mathsf{VK}}^{\mathrm{T}} \cdot \begin{bmatrix} \boldsymbol{c}_{\mathsf{rec}}^{(1)} \\ \boldsymbol{c}_{\mathsf{rec}}^{(2)} \end{bmatrix} \right) \Big/ \left\lfloor \frac{q}{2} \right\rceil \right\rceil, \tag{12}$$

and return the obtained $\boldsymbol{m} \in \{0,1\}^m$.

• $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}, \mathbf{reg}, \Psi, L)$: Decrypt the ciphertext $\boldsymbol{c}_{\mathsf{oa}}^{(1)} = (\boldsymbol{c}_{\mathsf{oa}}^{(1,1)}, \boldsymbol{c}_{\mathsf{oa}}^{(1,2)})$ by preforming the following steps:

(1) Check $\mathsf{Ver}(\mathsf{VK}, \Sigma, (\boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, L))$, return $\perp$ if the value is 0. Otherwise, parse the secret key $\mathsf{sk}_{\mathsf{OA}}$ as $\boldsymbol{S}_1 \in {\mathbb{Z}_q}^{n \times \ell}$ and the ciphertext $\boldsymbol{c}_{\mathsf{oa}}$ as in (10).

(2) Compute $(j'_1, \ldots, j'_\ell) = \lfloor (\boldsymbol{c}_{\mathsf{oa}}^{(1,2)} - \boldsymbol{S}_1^{\mathrm{T}} \cdot \boldsymbol{c}_{\mathsf{oa}}^{(1,1)}) / (p/2) \rceil \in \{0,1\}^\ell$, and output an integer $j \in [0, N-1]$ if the integer has the freshly computed binary representation, and the group information info contains $w^{(j)}$ with $\mathbf{reg}[j] \neq \mathbf{0}^{nk_1}$.

• $\langle \mathcal{P}, \mathcal{V} \rangle$: Both parties take gpk, $(\boldsymbol{A}_R, \boldsymbol{u}_R)$ and the ciphertext Ψ as the public input, and the prover gets the set of a message $\boldsymbol{m} \in \{0,1\}^{m_2}, \mathsf{pk}_j = \boldsymbol{B}_j, \mathsf{cert}_j$, and the random coins coins_Ψ used to generate Ψ as input, as specified above.

The goal of the prover is to carry out a ZK proof to convince the verifier that the secret inputs he made satisfy the following:

(1) $\boldsymbol{A}_R \cdot \boldsymbol{m} = \boldsymbol{u}_R \bmod q$.

(2) $\boldsymbol{G}_1 \cdot \boldsymbol{p}_j = \boldsymbol{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\boldsymbol{B}_j) \bmod p$ and $\mathsf{TVerify}_{\boldsymbol{A}}(\boldsymbol{u}, \boldsymbol{p}_j, w^{(j)}) = 1$.

(3) Vectors $\boldsymbol{x}_{\mathsf{rec}}, \boldsymbol{y}_{\mathsf{rec}}$ have infinity B-bounded norms, $\boldsymbol{z}_{\mathsf{rec}}$ has $\beta m_2 B$-bounded norm, as well as $\boldsymbol{r}_{\mathsf{oa}}^{(i)}$ lives in $\{0,1\}^{\widetilde{m}}$ for each $i = 1, 2$.

(4) Eqs. (9) and (10) hold.

In order to achieve this aim, $\mathcal{P}$ performs the following executions:

(1) Perform decomposition on matrices and vectors, respectively. Map the matrix $\boldsymbol{B}_j \in \mathbb{Z}_q^{n \times \bar{m}}$ into $\boldsymbol{b}_j = \mathsf{mdec}_{n,\bar{m},q}(\boldsymbol{B}_j^{\mathrm{T}}) \in \{0,1\}^{n\bar{m}k_2}$, and the vector $\boldsymbol{s}_{\mathsf{rec}}$ into $\boldsymbol{s}_{0,\mathsf{rec}} = \mathsf{vdec}_{n,q-1}(\boldsymbol{s}_{\mathsf{rec}}) \in \{0,1\}^{nk_2}$, resulting into $\boldsymbol{z}_\Psi = \mathsf{expand}^{\otimes}(\boldsymbol{b}_j, \boldsymbol{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k_2^2}$. Set

$$\boldsymbol{Q} = \boldsymbol{H}_{\bar{m},q-1} \cdot [\overbrace{\boldsymbol{Q}_0 | \cdots | \boldsymbol{Q}_0}^{n \text{ times}}] \in \mathbb{Z}_q^{\bar{m} \times 4n\bar{m}k_2^2}, \tag{13}$$

where the composite matrix $\boldsymbol{Q}_0 = \boldsymbol{I}_{\bar{m}k_2} \otimes \boldsymbol{g}' \in \mathbb{Z}_q^{\bar{m}k_2 \times 4n\bar{m}k_2^2}$ is shown as in Subsection 3.1.

(2) Produce a ZKAoK of the following set of relations:

$$\begin{cases} \boldsymbol{j} = (j_1, \ldots, j_\ell)^{\mathrm{T}} \in \{0,1\}^\ell, \\ (\boldsymbol{p}_j, (\boldsymbol{w}_\ell^{(j)}, \ldots, \boldsymbol{w}_1^{(j)})) \in \{0,1\}^{nk_1} \times (\{0,1\}^{nk_1})^\ell, \\ \boldsymbol{b}_j \in \{0,1\}^{n\bar{m}k_2},\ \boldsymbol{s}_{0,\mathsf{rec}} \in \{0,1\}^{nk_2}, \\ \boldsymbol{z}_\Psi = \mathsf{expand}^{\otimes}(\boldsymbol{b}_j, \boldsymbol{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k_2^2}, \\ \|\boldsymbol{x}_{\mathsf{rec}}\|_\infty, \|\boldsymbol{y}_{\mathsf{rec}}\|_\infty \leqslant B,\ \|\boldsymbol{z}_{\mathsf{rec}}\|_\infty \leqslant \beta m_2 B, \\ \boldsymbol{m} \in \{0,1\}^{m_2},\ \boldsymbol{r}_{\mathsf{oa}}^{(1)}, \boldsymbol{r}_{\mathsf{oa}}^{(2)} \in \{0,1\}^{\widetilde{m}}, \end{cases} \tag{14}$$

which demonstrates the following system of equations (in appropriate order) holds:

$$\begin{cases} \boldsymbol{A} \cdot \mathsf{enc}(j_1, \boldsymbol{v}_1) + \boldsymbol{A} \cdot \mathsf{enc}(\bar{j}_1, \boldsymbol{w}_1) = \boldsymbol{G}_1 \cdot \boldsymbol{u} \bmod p, \\ i \in [\ell - 1] : \boldsymbol{A} \cdot \mathsf{enc}(j_{i+1}, \boldsymbol{v}_{i+1}) + \boldsymbol{A} \cdot \mathsf{enc}(\bar{j}_{i+1}, \boldsymbol{w}_{i+1}) = \boldsymbol{G}_1 \cdot \boldsymbol{v}_i \bmod p, \\ \boldsymbol{0} = \boldsymbol{G}_1 \cdot \boldsymbol{p}_j + (-\boldsymbol{F}) \cdot \boldsymbol{b}_j \bmod p, \boldsymbol{v}_\ell = \boldsymbol{p}_j, \\ \boldsymbol{c}_{1,1} = \boldsymbol{B} \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)} \bmod p, \boldsymbol{c}_{1,2} = \boldsymbol{P}_1 \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)} + \left(\left\lfloor \frac{p}{2} \right\rceil \cdot \boldsymbol{I}_\ell \right) \cdot \boldsymbol{j} \bmod p, \\ \boldsymbol{c}_{2,1} = \boldsymbol{B} \cdot \boldsymbol{r}_{\mathsf{oa}}^{(2)} \bmod p, \boldsymbol{c}_{2,2} = \boldsymbol{P}_2 \cdot \boldsymbol{r}_{\mathsf{oa}}^{(2)} + \left(\left\lfloor \frac{p}{2} \right\rceil \cdot \boldsymbol{I}_\ell \right) \cdot \boldsymbol{j} \bmod p, \\ \boldsymbol{c}_{\mathsf{rec}}^{(1)} = (\bar{\boldsymbol{A}}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}) \cdot \boldsymbol{s}_{0,\mathsf{rec}} + \boldsymbol{I}_{m_2} \cdot \boldsymbol{y}_{\mathsf{rec}} \bmod q, \\ \boldsymbol{c}_{\mathsf{rec}}^{(2)} = \boldsymbol{Q} \cdot \boldsymbol{z}_\Psi + (\boldsymbol{G}_2^{\mathrm{T}} \cdot \boldsymbol{H}_{\mathsf{VK}}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}) \cdot \boldsymbol{s}_{0,\mathsf{rec}} + \boldsymbol{I}_{\bar{m}} \cdot \boldsymbol{z}_{\mathsf{rec}} \bmod q, \\ \boldsymbol{c}_{\mathsf{rec}}^{(3)} = (\boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}) \cdot \boldsymbol{s}_{0,\mathsf{rec}} + \boldsymbol{I}_{m_2} \cdot \boldsymbol{x}_{\mathsf{rec}} + \left(\left\lfloor \frac{q}{2} \right\rceil \cdot \boldsymbol{I}_{m_2} \right) \cdot \boldsymbol{m} \bmod q, \\ \boldsymbol{u}_R = \boldsymbol{A}_R \cdot \boldsymbol{m} \bmod q. \end{cases} \tag{15}$$

Let $\boldsymbol{t}_1 = \boldsymbol{b}_j$, $\boldsymbol{t}_2 = \boldsymbol{s}_{0,\mathsf{rec}}$, $\boldsymbol{t}_3 = \mathsf{expand}^{\otimes}(\boldsymbol{t}_1, \boldsymbol{t}_2)$, $\boldsymbol{t}_4 = (\mathsf{enc}^{\mathrm{T}}(j_1, \boldsymbol{v}_1) \| \mathsf{enc}^{\mathrm{T}}(\bar{j}_1, \boldsymbol{w}_1) \| \cdots \| \mathsf{enc}^{\mathrm{T}}(j_\ell, \boldsymbol{p}_j) \| \mathsf{enc}^{\mathrm{T}}(\bar{j}_\ell, \boldsymbol{w}_\ell))^{\mathrm{T}}$, $\boldsymbol{t}_5 = (\boldsymbol{u}^{\mathrm{T}} \| \boldsymbol{v}_1^{\mathrm{T}} \| \cdots \| \boldsymbol{v}_{\ell-1}^{\mathrm{T}})^{\mathrm{T}}$, $\boldsymbol{t}_6 = \boldsymbol{p}_j$, $\boldsymbol{t}_7 = \boldsymbol{m}$, $\boldsymbol{t}_8 = \boldsymbol{j}$, $\boldsymbol{t}_9 = \boldsymbol{r}_{\mathsf{oa}}^{(1)}$, $\boldsymbol{t}_{10} = \boldsymbol{r}_{\mathsf{oa}}^{(2)}$, $\boldsymbol{t}_{11} = \boldsymbol{x}_{\mathsf{rec}}$, $\boldsymbol{t}_{12} = \boldsymbol{y}_{\mathsf{rec}}$, $\boldsymbol{t}_{13} = \boldsymbol{z}_{\mathsf{rec}}$. Accordingly, set $\boldsymbol{M}_{1,4} = \boldsymbol{I}_\ell \otimes [\boldsymbol{A} | \boldsymbol{A}]$, $\boldsymbol{M}_{1,5} = \boldsymbol{I}_\ell \otimes (-\boldsymbol{G}_1)$, $\boldsymbol{M}_{2,1} = -\boldsymbol{F}$, $\boldsymbol{M}_{2,6} = \boldsymbol{G}_1$, $\boldsymbol{M}_{3,9} = \boldsymbol{M}_{5,10} = \boldsymbol{B}$, $\boldsymbol{M}_{4,8} = \boldsymbol{M}_{6,8} = \lfloor \frac{p}{2} \rceil \cdot \boldsymbol{I}_\ell$, $\boldsymbol{M}_{4,9} = \boldsymbol{P}_1$, $\boldsymbol{M}_{6,10} = \boldsymbol{P}_2$, $\boldsymbol{M}_{7,2} = \bar{\boldsymbol{A}}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}$, $\boldsymbol{M}_{7,12} = \boldsymbol{M}_{9,11} = \boldsymbol{I}_{m_2}$, $\boldsymbol{M}_{8,2} = \boldsymbol{G}_2^{\mathrm{T}} \cdot \boldsymbol{H}_{\mathsf{VK}}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}$, $\boldsymbol{M}_{8,3} = \boldsymbol{Q}$, $\boldsymbol{M}_{8,13} = \boldsymbol{I}_{\bar{m}}$, $\boldsymbol{M}_{9,2} = \boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{H}_{n,q-1}$, $\boldsymbol{M}_{9,7} = \lfloor \frac{q}{2} \rceil \cdot \boldsymbol{I}_{m_2}$, $\boldsymbol{M}_{10,7} = \boldsymbol{A}_R$, and $\boldsymbol{u}_3 = \boldsymbol{c}_{1,1}$, $\boldsymbol{u}_4 = \boldsymbol{c}_{1,2}$, $\boldsymbol{u}_5 = \boldsymbol{c}_{2,1}$, $\boldsymbol{u}_6 = \boldsymbol{c}_{2,2}$, $\boldsymbol{u}_7 = \boldsymbol{c}_{\mathsf{rec}}^{(1)}$, $\boldsymbol{u}_8 = \boldsymbol{c}_{\mathsf{rec}}^{(2)}$, $\boldsymbol{u}_9 = \boldsymbol{c}_{\mathsf{rec}}^{(3)}$, $\boldsymbol{u}_{10} = \boldsymbol{u}_R$ and also set the left public matrices $\{\boldsymbol{M}_{i,j}\}$ and vectors $\{\boldsymbol{u}_i\}$ as being zero, which allows writing the

system (15) as:

$$\begin{cases} \boldsymbol{u}_1 = \boldsymbol{M}_{1,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{1,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{1,13} \cdot \boldsymbol{t}_{13} \bmod p, \\ \quad \vdots \\ \boldsymbol{u}_6 = \boldsymbol{M}_{6,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{6,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{6,13} \cdot \boldsymbol{t}_{13} \bmod p, \\ \boldsymbol{u}_7 = \boldsymbol{M}_{7,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{7,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{7,13} \cdot \boldsymbol{t}_{13} \bmod q, \\ \quad \vdots \\ \boldsymbol{u}_{10} = \boldsymbol{M}_{10,1} \cdot \boldsymbol{t}_1 + \boldsymbol{M}_{10,2} \cdot \boldsymbol{t}_2 + \cdots + \boldsymbol{M}_{10,13} \cdot \boldsymbol{t}_{13} \bmod q. \end{cases} \tag{16}$$

We note that the present argument system is achieved by performing the protocol constructed in Subsection 3.2, whose negligibly soundness error is obtained after repeating it κ times.

Correctness. The correctness of the proposed group encryption follows from correctly decrypting the ABB ciphertext and the ordinary LWE ciphertext, which may cause some decryption errors. To decrypt the ciphertext $\mathsf{c}_{\mathsf{rec}}$, by utilizing procedure $\mathsf{DEC}(\mathsf{sk}_j, \Psi, L)$, we obtain

$$\boldsymbol{c}_{\mathsf{rec}}^{(3)} - \boldsymbol{E}_{\mathsf{VK}}^{\mathrm{T}} \cdot \begin{bmatrix} \boldsymbol{c}_{\mathsf{rec}}^{(1)} \\ \boldsymbol{c}_{\mathsf{rec}}^{(2)} \end{bmatrix} = \boldsymbol{x}_{\mathsf{rec}} - \boldsymbol{E}_{\mathsf{VK}}^{\mathrm{T}} \cdot \begin{bmatrix} \boldsymbol{y}_{\mathsf{rec}} \\ \boldsymbol{z}_{\mathsf{rec}} \end{bmatrix} + \boldsymbol{m} \cdot \left\lfloor \frac{q}{2} \right\rceil. \tag{17}$$

Note that $\|\boldsymbol{x}_{\mathsf{rec}}\|_\infty$ and $\|\boldsymbol{y}_{\mathsf{rec}}\|_\infty$ both have upper bound B, resulting $\|\boldsymbol{z}_{\mathsf{rec}}\|_\infty = \|\boldsymbol{R}_{\mathsf{rec}}^{\mathrm{T}} \cdot \boldsymbol{y}_{\mathsf{rec}}\|_\infty \leqslant \beta m_2 B = \widetilde{\mathcal{O}}(n^2)$. In addition, observing that the matrix $\boldsymbol{E}_{\mathsf{VK}}$ is generated by using Gaussian preimage sample technique, it will be similarly considered due to its elements bounded by $\widetilde{\mathcal{O}}(\sqrt{n})$. Therefore, the norm of error term of (17) will be not beyond $\widetilde{\mathcal{O}}(n^{3.5})$, which shows that the discussed error can be negligible when divided by $\lfloor \frac{q}{2} \rceil$ and performed with rounding operation. All of this ensures that the decryption algorithm will be exactly performed and finally return the desired $\boldsymbol{m}$ with high probability. This gives the correctness of $\mathsf{DEC}(\mathsf{sk}_j, \Psi, L)$.

For $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}, \mathbf{reg}, \Psi, L)$, a similar analysis is performed, where we only need to consider the case $i = 1$ as follows:

$$\begin{aligned} \boldsymbol{c}_{\mathsf{oa}}^{(1,2)} - \boldsymbol{S}_1^{\mathrm{T}} \cdot \boldsymbol{c}_{\mathsf{oa}}^{(1,1)} &= \left(\boldsymbol{S}_1^{\mathrm{T}} \cdot \boldsymbol{B} + \boldsymbol{E}_1\right) \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)} + \left\lceil \frac{p}{2} \right\rfloor \cdot \boldsymbol{j} - \boldsymbol{S}_1^{\mathrm{T}} \cdot \boldsymbol{B} \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)} \\ &= \boldsymbol{E}_1 \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)} + \left\lceil \frac{p}{2} \right\rfloor \cdot \boldsymbol{j}. \end{aligned} \tag{18}$$

Note that, for the specific parameters setting, the decryption procedure will return $\boldsymbol{j} = (j_1, \ldots, j_\ell)^{\mathrm{T}}$ if $\|\boldsymbol{E}_1 \cdot \boldsymbol{r}_{\mathsf{oa}}^{(1)}\|_\infty < p/4$ with overwhelming probability. This closes the desired correctness.

In the end, we argue that, since our argument system constructed in Subsection 3.2 has the perfect completeness, if a certified group member genuinely executes all prescribed algorithms, then he can compute the valid witness-vectors that are applicable in the proof protocol to generate accepted proofs by the verifier.

4.2 Efficiency analysis

It is observed that the implementations of the presented group encryption can be completed in polynomial time since all the algorithms involved are polynomially effective. For clarity, evaluations for bit-sizes of keys, ciphertexts, and communication cost of protocol between prover and verifier, will be given as follows.

– The size of public key of GM is dominated by the one-time signature scheme it chooses, and bit-size $\widetilde{\mathcal{O}}(\lambda^2)$ is available. The public key of OA consists of a set of matrices having total bit-size $\widetilde{\mathcal{O}}(\lambda^2 + \ell^2\lambda)$, and user's public key is provided with a matrix of size $\widetilde{\mathcal{O}}(\lambda^2)$.

– The secret keys of GM and users are given by bit string and trapdoors of bit-size $\widetilde{\mathcal{O}}(\lambda^2)$, along with cert of size $\widetilde{\mathcal{O}}(\ell\lambda)$, while OA's secret key is shown with $\widetilde{\mathcal{O}}(\ell^2\lambda)$ bits.

– The generated ciphertext Ψ contains three components, i.e., one ABB ciphertext of size $(2m_2 + \bar{m})\lceil \log q \rceil$, and one ordinary LWE ciphertext of size $2(n+\ell)\lceil \log p \rceil$ as well as a one-time signature Σ for the first two components, resulting the total bit-size $\widetilde{\mathcal{O}}(\lambda) + |\Sigma|$.

– The communication cost of the ZKAoK heavily relies on the size of $\boldsymbol{z}_\Psi = \mathsf{expand} \otimes (\boldsymbol{b}_j, \mathsf{s}_{0,j}) \in \{0,1\}^{4n\bar{m}k_2^2}$ and its bit-size is quantized as $\widetilde{\mathcal{O}}(\lambda^2)$.

Table 1 Comparison between scheme [4] and ours[a)]

Scheme	GM PK	GM SK	OA PK	OA SK	User's PK	User's SK	Ciphertext	Communication
LLMNW [4]	68.60 GB	482.55 GB	2.37 GB	38.86 GB	2.37 GB	38.86 GB	$\|\Sigma\|$+8.67 MB	3728.00 TB
Ours	71.75 MB	148.49 MB	74.12 MB	21.18 KB	192.38 MB	2.67 GB	$\|\Sigma'\|$+1.06 MB	255.81 TB

a) We remark that, in the above, $|\Sigma| = 2.36$ TB and $|\Sigma'| = 82.99$ GB, which directly follows from the use of one-way function of the form $\boldsymbol{M} \cdot \boldsymbol{x} = \boldsymbol{y} \bmod q$ in $\mathcal{OTS}$ scheme, where $\boldsymbol{M} \in U(\mathbb{Z}_q^{n\times m})$, $\boldsymbol{x} \in \{0,1\}^m$, and $m = n\lceil \log q \rceil$.

To better understand the efficiency advantage, we also provide a simple comparison between our scheme and the LLMNW scheme [4], which is the only currently available group encryption instantiation over lattices. We consider 80-bit security and a 2^{-80} soundness error (implying that the repetition number of the protocol described in Subsection 3.2 is specified as $\kappa = 137$) under the same group size $N = 2^{10}$. We adopt the same approach as in [39–41] to estimate the security of the schemes. Following the specific routine, we run the BKZ cost estimator[2)] with lattice sieving algorithms [42, 43] (which are expected to be significantly faster than enumeration algorithms [44, 45]), to obtain the root Hermite factor (RHF) 1.0048 corresponding to 80-bit security. Then, by the relations between RHF and lattice problems shown in [39], we adopt suitable parameters as: $(n, q) = (2795, 1125899906842679 \approx 2^{50})$ in scheme [4] and $(n, p, q) = (913, 262147 \approx 2^{18}, 8796093022237 \approx 2^{43})$ in this work[3)]. The results are summarized in Table 1.

It is evident that our proposed scheme is essentially comparable to scheme [4]. Although our scheme represents dramatic improvements, it is still far away from practical implementation. Cryptographic components with trapdoors are always inefficient due to the excessive limitations on the chosen parameters. We note that the GE will obtain dramatic improvements in terms of efficiency when its design does not rely on any trapdoor. The most significant advantage of our scheme is to reduce the use of lattice trapdoors, which consequently mitigates the efficiency dilemma in the sense of heavy trapdoors.

4.3 Security analysis

We provide positive provable security analysis for our scheme under SIS and LWE hardness assumptions with the help of classical reduction methods as follows.

Theorem 2. Suppose that the Stern-like argument systems shown in Subsection 3.2 are simulation-sound and the one-time signature $\mathcal{OTS}$ possesses the strong unforgeability. Then, the scheme provides anonymity and message secrecy under the $\mathsf{LWE}_{n,q,\chi}$ assumption.

Proof. We provide the proofs of Lemmas 3 and 4 in the following, respectively, which together construct our desired proof for this theorem.

Lemma 3. The scheme provides anonymity if the $\mathsf{LWE}_{n,q,\chi}$ assumption holds and the one-time signature $\mathcal{OTS}$ is strongly unforgeable.

Proof. The proof is similar to [46] and is proceeded via the following sequence of games in which the first one conducts the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon\text{-}0}}(\lambda)$ and the final one is executed as the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon\text{-}1}}(\lambda)$. By demonstrating that any two consecutive games are indistinguishable, this proof will be completed. For simplicity, hereunder we take PPT algorithms $\mathcal{A}$ and $\mathcal{B}$ as the adversary and challenger, respectively, and denote by W_i the result of the adversary in game i.

Game 1. This is exactly the game $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon\text{-}0}}(\lambda)$ where after the challenger $\mathcal{B}$ publicizes the parameters pp, the opening authority OA sends its public key $\mathsf{pk}_{\mathsf{OA}} = (\boldsymbol{B}, \boldsymbol{P}_1, \boldsymbol{P}_2)$ to $\mathcal{A}$ who introduces the group members into the group on behalf of the GM via access to the USER oracle.

Specially, $\mathcal{A}$ selects two registered public keys $\mathsf{pk}_{\mathsf{U},0}, \mathsf{pk}_{\mathsf{U},1} \in \mathbb{Z}_q^{n\times \bar{m}}$ given by the challenger, and outputs $((\boldsymbol{A}_R, \boldsymbol{u}_R), \boldsymbol{m}, L)$ for which $\boldsymbol{u}_R = \boldsymbol{A}_R \cdot \boldsymbol{m} \bmod q$, with $\boldsymbol{A}_R \in \mathbb{Z}_q^{n\times m}$, $\boldsymbol{u}_R \in \mathbb{Z}_q^n$, and $\boldsymbol{m} \in \{0,1\}^m$. Later on, the challenger takes $b \hookleftarrow \{0,1\}$ and computes a challenge ciphertext $\Psi^* = (\mathsf{VK}^*, \boldsymbol{c}^*_{\mathsf{rec}}, \boldsymbol{c}^*_{\mathsf{oa}}, \Sigma^*)$ of the message $\boldsymbol{m}$ under $\mathsf{pk}_{\mathsf{U},b} = \boldsymbol{B}_{\mathsf{U},b}$. Then, generate the corresponding proof $\pi^*_{\Psi^*}$ and execute the prescribed queries and responses. When $\mathcal{A}$ halts, it returns the result $b' \in \{0,1\}$, which gives the success probability $\Pr[\mathbf{Exp}_{\mathsf{A}}^{\mathsf{anon\text{-}0}}(\lambda) = 1]$.

Game 2. The challenger aborts this experiment when $\mathcal{A}$ makes the opening query to the ciphertext $\Psi = (\mathsf{VK}, \boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, \Sigma)$ such that $\mathsf{VK} = \mathsf{VK}^*$ and Σ is valid (assuming that VK^* is generated in advance).

2) Schanck J. Estimator. https://github.com/jschanck/estimator.

3) Here we take two suitable moduli for our scheme rather than only using a modulus. This action follows from the observation that using a mechanism that does not involve trapdoors facilitates a more efficient choice of parameters.

At this case, it requires that the security of $\mathcal{OTS}$ is broken by $\mathcal{A}$, which means that there exists an efficient forger $\mathcal{B}$ for $\mathcal{OTS}$ such that $|\Pr[W_2] - \Pr[W_1]| \leqslant \mathbf{Adv}^{\mathsf{ots}}(\lambda)$. This leads that, assuming $\mathcal{OTS}$ having strong unforgeability, the Gmae 2 is like Game 1.

Game 3. This game is the same as Game 2 except for one modification that adds $\boldsymbol{S}_2$ to $\mathsf{sk}_{\mathsf{OA}}$. It makes no difference in the view of $\mathcal{A}$, leading $\Pr[W_3] = \Pr[W_2]$.

Game 4. This game modifies the oracle query of OPEN by using $\boldsymbol{S}_2$ to substitute $\boldsymbol{S}_1$. At this point, the view of $\mathcal{A}$ is identical to in Game 3 except for the event F_1, where a query to the OPEN oracle for a valid ciphertext $\Psi = (\mathsf{VK}, \boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, \Sigma)$ with $\boldsymbol{c}_{\mathsf{oa}} = (\boldsymbol{c}_{\mathsf{oa}}^{(1)}, \boldsymbol{c}_{\mathsf{oa}}^{(2)})$ encrypting distinct bit strings, happens. However, this clearly means that the soundness of the ZKAoK considered in Subsection 3.2 is broken, giving $|\Pr[W_4] - \Pr[W_3]| \leqslant \Pr[F_1] \leqslant \mathbf{Adv}^{\mathsf{sound}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 5. Here we modify Game 4 in the generation of proofs $\pi^*_{\Psi^*}$ via appealing the simulator of the ZKAoK of Subsection 3.2, instead of computing proofs by the real witnesses $\boldsymbol{r}_{\mathsf{oa}}^{(1)}, \boldsymbol{r}_{\mathsf{oa}}^{(2)}$. Note that, assuming that public parameters are generated in a trust manner, simulated proofs trivially follow from the statistical zero-knowledge simulator (employing, e.g., Damgård's technology [47]). Then, by the statistical zero-knowledgeness of the ZKAoK, this game is statistically indistinguishable to Game 4 in the $\mathcal{A}$'s view, which leads $\Pr[W_5] = \Pr[W_4]$.

Game 6. This game has a slight modification in $\boldsymbol{c}_{\mathsf{oa}}$ by taking $\boldsymbol{c}_{\mathsf{oa}}^{(1)}$ as the encryption of $\mathsf{bin}(j_1)$ while pertaining $\boldsymbol{c}_{\mathsf{oa}}^{(2)}$ still for $\mathsf{bin}(j_0)$ for the challenge ciphertext Ψ^*. Note that this change is negligible to $\mathcal{A}$ since the Regev encryption shares the semantic security for public key $(\boldsymbol{B}, \boldsymbol{P}_1)$, and it further incurs $|\Pr[W_6 = 1] - \Pr[W_5 = 1]| = \mathsf{negl}(\lambda)$ by the fact that the OPEN queries are proceeded by $\boldsymbol{S}_2$.

Game 7. This game makes one change by switching back to the application of $\boldsymbol{S}_1$ for the OPEN queries with discarding $\boldsymbol{S}_2$, and the modification is invariant to the adversary except the event F_2, where the queries to the OPEN for a valid ciphertext Ψ containing $\boldsymbol{c}_{\mathsf{oa}}^{(1)}, \boldsymbol{c}_{\mathsf{oa}}^{(2)}$ encrypting distinct identities, happens. But, the occurrence of F_2 implies that the simulation soundness of the underlying ZKAoK system used to generate Π_{GE} is broken. This results into $|\Pr[W_7 = 1] - \Pr[W_6 = 1]| \leqslant \mathbf{Adv}^{\mathsf{ss}}_{\Pi_{\mathsf{GE}}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 8. Here, this experiment performs a modification to the Game 7 only by taking $\boldsymbol{c}_2$ as the encryption of $\mathsf{bin}(j_1)$ for the challenge ciphertext Ψ^*. Note that this change is unnoticed to $\mathcal{A}$ due to the semantic security the encryption shares for public key $(\boldsymbol{B}, \boldsymbol{P}_2)$, and also for the application of $\boldsymbol{S}_1$ to the OPEN, we have $|\Pr[W_8 = 1] - \Pr[W_7 = 1]| = \mathsf{negl}(\lambda)$.

Game 9. Here, this experiment generates a real proof for ciphertext Ψ^* instead of using simulated proof, which is the only modification different to Game 8. The statistical zero-knowledgeness of the underlying ZKAoK system makes the difference between Game 8 and Game 9 negligible, i.e., $\Pr[W_8 = 1] \approx \Pr[W_9 = 1]$. This is actually the experiment $\mathbf{Exp}^{\mathsf{anon\text{-}1}}_{\mathcal{A}}(\lambda)$, which directly leads that $\Pr[W_9 = 1] = \mathbf{Exp}^{\mathsf{anon\text{-}1}}_{\mathcal{A}}(\lambda)$. By these above games, we have

$$|\mathbf{Exp}^{\mathsf{anon\text{-}1}}_{\mathcal{A}}(\lambda) - \mathbf{Exp}^{\mathsf{anon\text{-}0}}_{\mathcal{A}}(\lambda)| = \mathsf{negl}(\lambda). \tag{19}$$

This provides anonymity.

Lemma 4. The scheme provides message secrecy if the $\mathsf{LWE}_{n,q,\chi}$ assumption holds and the one-time signature $\mathcal{OTS}$ is strongly unforgeable.

Proof. This proof is similar to Theorem 3 of [4] in that similar security definition and encryption mechanism (i.e., ABB scheme [38]) are employed. We complete it by using a sequence of statistically indistinguishable games where the first game is exactly the real experiment in (i.e., the case in which the challenger chooses the bit $b = 1$, and the ciphertext and corresponding proof are generated from the real encryption and the $\mathsf{PROVE}(\cdot)$ oracle). Whereas, the last game is the simulated experiment where the encryption is performed for some random plaintext and the associated proofs are produced by a simulator $\mathcal{P}'$. For simplicity, we denote the event that the adversary $\mathcal{A}$ returns $b' = 1$ by W_i in game i.

Game 1. By the security model, in this game, the public parameters pp including $\bar{\boldsymbol{A}}, \boldsymbol{U} \in \mathbb{Z}_q^{n \times m}$ are fed to $\mathcal{A}$ by the challenger. Then the adversary generates public keys $\mathsf{pk}_{\mathsf{OA}} = (\boldsymbol{B}, \boldsymbol{P}_1, \boldsymbol{P}_2) \in \mathbb{Z}_q^{n \times \widetilde{m}} \times (\mathbb{Z}_q^{\ell \times \widetilde{m}})^2$ and $\mathsf{pk}_{\mathsf{GM}}$, which means that both the OA and the GM are in its whole control, and it can execute operations on behalf of them. The JOIN protocol between the challenger and $\mathcal{A}$ is run to register the certified public key $\mathsf{pk}_{\mathsf{U}} = \boldsymbol{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ which is chosen by challenger for some genuine receiver. After that, the adversary $\mathcal{A}$ makes a polynomial number of queries to $\mathsf{DEC}(\cdot)$ oracle, and the challenger faithfully handles by the private key $\mathsf{sk}_{\boldsymbol{B}} = \boldsymbol{T}_{\boldsymbol{B}} \in \mathbb{Z}^{\bar{m} \times m}$ satisfying $\boldsymbol{B}_{\mathsf{U}} \cdot \boldsymbol{T}_{\boldsymbol{B}} = \boldsymbol{0}^{n \times m}$. Then, $\mathcal{A}$ provides a triple $((\boldsymbol{A}_R, \boldsymbol{u}_R), \boldsymbol{m}, L)$ satisfying $\boldsymbol{u}_R = \boldsymbol{A}_R \cdot \boldsymbol{m}$ with $\boldsymbol{A}_R \in \mathbb{Z}_q^{n \times m}$, $\boldsymbol{u}_R \in \mathbb{Z}_q^n$, and $\boldsymbol{m} \in \{0,1\}^m$. Later on, the

challenger computes $\Psi^* = (\mathsf{VK}^*, \boldsymbol{c}^*_{\mathsf{rec}}, \boldsymbol{c}^*_{\mathsf{oa}}, \Sigma^*)$ for the real message $\boldsymbol{m}$ using $\mathsf{pk}_{\mathsf{U}} = \boldsymbol{B}_{\mathsf{U}}$ as the encryption key, as well as a polynomial number of proofs $\pi^*_{\Psi^*}$ associated with this ciphertext to $\mathcal{A}$. Then, $\mathcal{A}$ is given the access to the $\mathsf{DEC}(\cdot)$ oracle with the prescribed restrictions. When the game is over, it returns a bit $b' \in \{0, 1\}$.

Game 2. This experiment imposes some modification on the $\mathsf{DEC}(\cdot)$ oracle to reject any ciphertext $\Psi = (\mathsf{VK}, \boldsymbol{c}_{\mathsf{rec}}, \boldsymbol{c}_{\mathsf{oa}}, \Sigma)$ which satisfies $\mathsf{VK} = \mathsf{VK}^*$ (here, we assume that VK^* can be produced before this game w.l.o.g.). It can be seen that Game 2 is like Game 1 except that it denies the ciphertext that is accepted in Game 1. We note that it can only occur when $\mathcal{A}$ breaks the $\mathcal{OTS}$ that has strong unforgeability. Thus, we have $|\Pr[W_2] - \Pr[W_1]| \leqslant \mathbf{Adv}^{\mathrm{ots}}(\lambda)$, which is negligible under the assumption that $\mathcal{OTS}$ is strongly unforgeable.

Game 3. This experiment modifies the generation of proofs $\pi^*_{\Psi^*}$. Namely, we apply the zero-knowledge simulator presented in Subsection 3.2 rather than the witness used for Ψ^* to compute proof at each access to $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}$ (note that, given the trusted choice of parameters, the statistically perfect simulation can be achieved by the techniques [47] without requiring considerable rounds). Here, the statistical zero-knowledges ensures that the change is unnoticed, even if the adversary has an unbounded computation power: $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$. Hereunder, the PROVE oracle has never longer used the random coins $\mathrm{coins}^*_{\Psi} = (\boldsymbol{s}^*_{\mathsf{rec}}, \boldsymbol{R}^*_{\mathsf{rec}}, \boldsymbol{x}^*_{\mathsf{rec}}, \boldsymbol{y}^*_{\mathsf{rec}}, \boldsymbol{r}^{(1)*}_{\mathsf{oa}}, \boldsymbol{r}^{(2)*}_{\mathsf{oa}})$.

Game 4. In this game, we modify the generation of Ψ^* by encrypting a random element in $\mathbb{Z}_q^m$ as $\boldsymbol{c}^*_{\mathsf{rec}}$ where we no longer use the random coins as in Game 3. By the Lemma 9 in [4], it shows that any any noticed variation in $\mathcal{A}'$s view will imply the existence of a selective adversary that breaks the the ABB encryption. Further, by the theorem 25 of [38], we obtain that, based on the hardness of LWE problem, Game 4 is the same as Game 3: $|\Pr[W_4] - \Pr[W_3]| \leqslant \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

Game 5. Now we make a final change on the decryption oracle $\mathsf{DEC}(\cdot)$ and remove the rejection rule of Game 2. By the assumption that $\mathcal{OTS}$ is strongly unforgeable, we have $|\Pr[W_5] - \Pr[W_4]| \leqslant \mathbf{Adv}^{\mathsf{ots}}(\lambda) \leqslant \mathsf{negl}(\lambda)$. We note, in the last game, no any witness is known for the oracle $\mathsf{PROVE}(\cdot)$, which mirrors the experiment where bit $b = 0$ is chosen by challenger. Based on the above games, we have $|\Pr[W_5] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$. This completes the proof.

Put the above proofs together, we finally achieve the desired proof.

Theorem 3. The scheme is sound under the SIS assumption.

Proof. The proof is the same as that of [4] except for the differences of verifying group membership via lattice-based accumulators and computing the ciphertext $\boldsymbol{c}_{\mathsf{oa}}$ by the dual Regev encryption mechanism, and the procedure of proof directly follows the similar idea and is omitted here, where we additionally need to apply the property of SIS-based Merkle-tree accumulators to complete our result.

5 Conclusion

In this paper, we constructed a ZKAoK system that is both friendly to accumulated values and hidden matrices. Based on this system, we apply the lattice-based accumulators and the dual Regev encryption technique, neither of which involves lattice trapdoors, to propose a more efficient group encryption scheme over lattices than [4]. Our proposed scheme achieves drastic gains in efficiency while not suffering any loss in security.

Acknowledgements This work was supported by the National Cryptography Development Fund (Grant No. MMJJ20180110) and National Natural Science Foundation of China (Grant No. 61960206014).

References

1 Kiayias A, Tsiounis Y, Yung M. Group encryption. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Kuching, 2007. 181–199

2 Chaum D, Heyst E V. Group signatures. In: Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, 1991. 257–265

3 Trolin M, Wikström D. Hierarchical group signatures. In: Proceedings of International Colloquium on Automata, Languages, and Programming, Lisbon, 2005. 446–458

4 Libert B, Ling S, Mouhartem M, et al. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, 2016. 101–131

5 Regev O. On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, 2005. 84–93

6 Ajtai M. Generating hard instances of the short basis problem. In: Proceedings of International Colloquium on Automata, Languages, and Programming, Prague, 1999. 1–9

7 Libert B, Ling S, Mouhartem M, et al. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, 2016. 373–403

8 Lyubashevsky V. Lattice signatures without trapdoors. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, 2012. 738–755

9 Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, 2008. 197–206

10 Micciancio D, Peikert C. Trapdoors for lattices: simpler, tighter, faster, smaller. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, 2012. 700–718

11 Zhang J, Yu Y, Fan S Q, et al. Improved lattice-based CCA2-secure PKE in the standard model. Sci China Inf Sci, 2020, 63: 182101

12 Alwen J, Peikert C. Generating shorter bases for hard random lattices. In: Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, Freiburg, 2009. 75–86

13 Libert B, Ling S, Nguyen K, et al. Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, 2016. 1–31

14 Ling S, Nguyen K, Wang H X, et al. Lattice-based group signatures: achieving full dynamicity with ease. In: Proceedings of International Conference on Applied Cryptography and Network Security, Kanazawa, 2017. 293–312

15 Cash D, Hofheinz D, Kiltz E, et al. Bonsai trees, or how to delegate a lattice basis. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco, 2010. 523–552

16 Camenisch J, Lysyanskaya A. A signature scheme with efficient protocols. In: Proceedings of International Conference on Security in Communication Networks, Amalfi, 2002. 268–289

17 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Prague, 1999. 223–238

18 Cathalo J, Libert B, Yung M. Group encryption: non-interactive realization in the standard model. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, 2009. 179–196

19 Aimani L E, Joye M. Toward practical group encryption. In: Proceedings of the 11th International Conference on Applied Cryptography and Network Security, Banff, 2013. 237–252

20 Libert B, Yung M, Joye M, et al. Traceable group encryption. In: Proceedings of International Workshop on Public Key Cryptography, Buenos Aires, 2014. 592–610

21 Kiayias A, Tsiounis Y, Yung M. Traceable signatures. In: Proceedings of the 23rd Annual Eurocrypt Conference, Interlaken, 2004. 571–589

22 Izabachène M, Pointcheval D, Vergnaud D. Mediated traceable anonymous encryption. In: Proceedings of the 1st International Conference on Cryptology and Information Security in Latin America, Puebla, 2010. 40–60

23 Naor M, Yung M. Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, 1990. 427–437

24 Micciancio D, Peikert C. Hardness of SIS and LWE with small parameters. In: Proceedings of Annual Cryptology Conference, Santa Barbara, 2013. 21–39

25 Brakerski Z, Langlois A, Peikert C, et al. Classical hardness of learning with errors. In: Proceedings of A Symposium on Theory of Computing Conference, Palo Alto, 2013. 575–584

26 Peikert C. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, Bethesda, 2009. 333–342

27 Baric N, Pfitzmann B. Collision-free accumulators and fail-stop signature schemes without trees. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Konstanz, 1997. 480–494

28 Camenisch J, Lysyanskaya A. Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Proceedings of the 22nd Annual International Cryptology Conference, Santa Barbara, 2002. 61–76

29 Nguyen N. Accumulators from bilinear pairings and applications. In: Proceedings of Cryptographers' Track at the RSA Conference, San Francisco, 2005. 275–292

30 Tsudik G, Xu S H. Accumulating composites and improved group signing. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Taipei, 2003. 265–286

31 Stern J. A new paradigm for public key identification. IEEE Trans Inform Theory, 1996, 42: 1757–1768

32 Benhamouda F, Camenisch J, Krenn S, et al. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, 2014. 551–572

33 Jain A, Krenn S, Pietrzak K, et al. Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Beijing, 2012. 663–680

34 Langlois A, Ling S, Nguyen K, et al. Lattice-based group signature scheme with verifier-local revocation. In: Proceedings of International Workshop on Public Key Cryptography, Buenos Aires, 2014. 345–361

35 Ling S, Nguyen K, Stehlé D, et al. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Proceedings of International Workshop on Public Key Cryptography, Nara, 2013. 107–124

36 Ling S, Nguyen K, Wang H X. Group signatures from lattices: simpler, tighter, shorter, ring-based. In: Proceedings of IACR International Workshop on Public Key Cryptography, Gaithersburg, 2015. 427–449

37 Kawachi A, Tanaka K, Xagawa K. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, 2008. 372–389

38 Agrawal S, Boneh D, Boyen X. Efficient lattice (H)IBE in the standard model. In: Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco, 2010. 553–572

39 Yang R P, Au M H, Zhang Z F, et al. Efficient lattice-based zero-knowledge arguments with standard soundness: construction and applications. In: Proceedings of Annual International Cryptology Conference, 2019. 147–175

40 Albrecht M R, Player R, Scott S. On the concrete hardness of Learning with Errors. J Math Cryptology, 2015, 9: 169–203

41 Kosba A E, Zhao Z C, Miller A, et al. C∅C∅: a framework for building composable zero-knowledge proofs. Cryptology ePrint Archive, Report 2015/1093, 2005

42 Alkim E, Ducas L, Pöppelmann T, et al. Post-quantum key exchange — a new hope. In: Proceedings of the 25th USENIX Security Symposium, Austin, 2016. 327–343

43 Albrecht M R, Curtis R R, Deo A, et al. Estimate all the {LWE, NTRU} schemes! In: Proceedings of International Conference on Security and Cryptography for Networks, Amalfi, 2018. 351–367

44 Chen Y M, Nguyen P Q. BKZ 2.0: better lattice security estimates. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Seoul, 2011. 1–20

45 Zheng Z X, Wang X Y, Xu G W, et al. Orthogonalized lattice enumeration for solving SVP. Sci China Inf Sci, 2018, 61: 032115

46 Sahai A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, 1999. 543–553

47 Damgård I. Efficient concurrent zero-knowledge in the auxiliary string model. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, 2000. 418–430