Lattice-Based Group Encryption with Full Dynamicity and Message Filtering Policy

Jing Pan[1,2], Xiaofeng Chen[1,2](✉), Fangguo Zhang[3,4], and Willy Susilo[5]

[1] State Key Laboratory of Integrated Service Networks (ISN),
Xidian University, Xi'an 710071, China
jinglap@aliyun.com, xfchen@xidian.edu.cn

[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
isszhfg@mail.sysu.edu.cn

[4] Guangdong Province Key Laboratory of Information Security Technology,
Guangzhou 510006, China

[5] Institute of Cybersecurity and Cryptology, School of Computing and Information
Technology, University of Wollongong, Wollongong, NSW 2522, Australia
wsusilo@uow.edu.au

Abstract. Group encryption (GE) is a fundamental privacy-preserving primitive analog of group signatures, which allows users to decrypt specific ciphertexts while hiding themselves within a crowd. Since its first birth, numerous constructions have been proposed, among which the schemes separately constructed by Libert et al. (Asiacrypt 2016) over lattices and by Nguyen et al. (PKC 2021) over coding theory are post-quantum secure. Though the last scheme, at the first time, achieved the full dynamicity (allowing group users to join or leave the group in their ease) and message filtering policy, which greatly improved the state-of-affairs of GE systems, its practical applications are still limited due to the rather complicated design, inefficiency and the weaker security (secure in the random oracle model). In return, the Libert et al.'s scheme possesses a solid security (secure in the standard model), but it lacks the previous functions and still suffers from inefficiency because of extremely using lattice trapdoors. In this work, we re-formalize the model and security definitions of fully dynamic group encryption (FDGE) that are essentially equivalent to but more succinct than Nguyen et al.'s; Then, we provide a generic and efficient zero-knowledge proof method for proving that a binary vector is *non-zero* over lattices, on which a proof for the Prohibitive message filtering policy in the lattice setting is first achieved (yet in a simple manner); Finally, by combining appropriate cryptographic materials and our presented zero-knowledge proofs, we achieve the first lattice-based FDGE scheme in a simpler manner, which needs no any lattice trapdoor and is proved secure in the standard model (assuming interaction during the proof phase), outweighing the existing post-quantum secure GE systems in terms of functions, efficiency and security.

Keywords: Lattice cryptography · Group encryption · Full dynamicity · Message filtering · Zero-knowledge

M. Tibouchi and H. Wang (Eds.): ASIACRYPT 2021, LNCS 13093, pp. 156–186, 2021.
https://doi.org/10.1007/978-3-030-92068-5_6

1 Introduction

Group encryption (GE), introduced by Kiayias, Tsiounis and Yung (KTY) [21] as the natural encryption analog of group signature (GS) that was first conceptualized by Chaum and van Heyst [16], is a fundamental anonymity primitive that allows anonymizing valid decryptors within a population of certified users. Since the pioneering work [21], GE has found a wide range of applications (see, e.g., [21,25,35]) in filtering malformed encrypted emails, building oblivious retriever storage systems, trusted third parties as well as hierarchical group signatures [42]. Because of the duality, these two primitives share some common design ideas in offering user memberships and generating anonymous signatures/ciphertexts.

In the design of these two anonymity primitives, to build a group of certified users is a key component. In general, there are three types of groups optional for GS: The simplest choice is the static group [6], in which the group population is fixed at the setup phase and the public/private key pairs of group members are assigned by the group manager (GM) as memberships; The partially dynamic group [7,22,40] is then introduced to support dynamic and concurrent user enrollments but deny membership revocation. In such a group, a prospective user generates a key pair on his own, and then becomes a valid group member only when his application for joining the group is accepted by the GM, who computes a signature on user's public key and returns it back as the membership. Despite an essential functionality, support for membership revocation is quite challenging to realize in an efficient manner, since it requires that the signing algorithm is disabled for revoked users and no significant increase for workloads of other parties (i.e., managers, non-revoked users and verifiers) is seen. To address this problem, several approaches [8,11,12,36] have been suggested, resulting into the fully dynamic groups [9], where membership revocation is additionally allowed.

Unlike the context of group signature, the GE always uses the partially dynamic group in its design since its first formalization [21] for security reasons. This type of group allows prospective users dynamically and concurrently to join the group, but any valid application for revoking membership is rejected, which is quite unsatisfactory in the realistic world. In fact, group signatures with full dynamicity have attracted much attention and have been constructed both on pairing assumptions [28,33] and lattice assumptions [31]. To change this situation, in PKC 2021, Nguyen et al. [35] first considered the full dynamicity in the context of group encryption and proposed a code-based instantiation secure in the random oracle model. In their design, they also first considered the message filtering policies which are quite useful for practical applications of GE systems. However, their formalization of FDGE is adapted directly from that of fully dynamic group signature [9] and hard to understand. Moreover, the construction is rather complicated and inefficient even in the random oracle model. Therefore, it is encouraging to design a group encryption that captures the full dynamicity, message filtering policy and a solid security in a relatively simple manner.

Our Contributions. Motivated by the above discussion, we reconsider the full dynamicity in the context of group encryption, and propose a lattice-based instantiation in a simpler manner that shares the same functions as the existing FDGE scheme [35] and meanwhile outweighs all available post-quantum secure schemes [25,35] in terms of functions, efficiency and security. Our contributions are summarized as follows:

- By introducing appropriate ingredients into the KTY model that supports dynamic user enrollments but denies membership revocations, we re-formalize the model and security requirements of FDGE that are essentially equal to but more succinct and understandable than the currently existing model.
- We provide a generic and efficient zero-knowledge proof method for demonstrating that some binary vector is *non-zero* over lattices, on which we first achieve a lattice-based proof (also generic and efficient) for Prohibitive message filtering policy. Both proofs will serve for our subsequent construction.
- By making use of appropriate cryptographic materials and the presented zero-knowledge proofs, we achieve the first lattice-based group encryption secure in the standard model and with full dynamicity in a free-of-trapdoor manner, which meets our formalized model and outweighs all existing post-quantum secure GE schemes in terms of functions and efficiency.

Related Work. The privacy-preserving cryptography has been an extremely active research area in the last decades. As one of the fundamental anonymity primitives, group encryption thus has attracted noticeable attention in recent years. The relevant concepts and definitions were first introduced by Kiayias, Tsiounis and Yung [21], who also then put forth a modular design consisting of zero-knowledge proofs, digital signatures (e.g., [13]) and anonymous CCA2-secure public-key encryptions (e.g., [37]). Later, Cathalo et al. [15] designed a non-interactive scheme in the standard model for the goal of optimizing the number of rounds. Similarly, over weaker assumptions, Aimani et al. [1] proposed more practical schemes by utilizing succinct approaches to protect the identity of group members. For sake of balancing better privacy vs. safety, Libert et al. [29] supposed a variant with public traceability to specific ciphertexts, which was inspired from traceable signatures [20]. Further, to strengthen secrecy, Izabachène et al. [19] constructed traceable variants that are free of subliminal channels, stressing confidentiality, anonymity and traceability. However, all these instantiations are proposed over number-theoretic assumptions and are vulnerable under quantum attacks. This situation has been unchanged until Libert et al. [25] proposed the currently only existing lattice-based scheme recently.

What should be noted out is that, all the group encryptions discussed above only offer partial dynamicity that allows concurrent user enrollments but denies membership revocations, which is quite unsatisfactory in the most realistic applications. To end this situation, more currently, Nguyen et al. [35] proposed a fully dynamic group encryption scheme secure in the random oracle from coding theory, where they also achieved the message filtering policies. However, their model is directly adapted from that of fully dynamic group signature [9] and is tedious. Moreover, the proposed scheme is rather complicated and inefficient together with provable security in the random oracle model. This motivates us to construct a fully dynamic group encryption, in a simple manner, that share practical functions similar to the scheme [35] while obtaining high efficiency and solid security (against quantum attacks).

Organization. In the forthcoming sections, we briefly recall the needed lattice techniques and cryptographic blocks in Sect. 2. The formalized model of FDGE is given in Sect. 3. Section 4 describes our new techniques used for demonstrating inequalities of binary vectors and the underlying zero-knowledge argument system. In Sect. 5, we describe our scheme that captures all desired properties, of which analysis is given. Finally, Sect. 6 concludes our work.

2 Preliminaries

Notations. For any positive integers $n \geq k$, we denote the set $\{1, ..., n\}$ by $[n]$, the set $\{k, ..., n\}$ by $[k, n]$. All vectors are written as bold lower-case letters in the column form, and matrices as bold upper-case letters. For $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ with columns $(\mathbf{b}_i)_i$, their Euclidean l_2 norms are respectively written as $\|\mathbf{b}\|$ and $\|\mathbf{B}\| = \max_{i \leq m} \|\mathbf{b}_i\|$. If a given set $\mathcal{S}$ is finite, we let $U(\mathcal{S})$ to denote the uniform distribution over it and use $x \hookleftarrow D$ to represent the sampling action according to the distribution D. For two same-size binary vectors $\mathbf{x}$ and $\mathbf{y}$, we use $d_H(\mathbf{x}, \mathbf{y})$ to denote their Hamming distance, which is equal to l_1 norm $\|\mathbf{x} \oplus \mathbf{y}\|_1$.

2.1 Lattices and Computational Problems

As in [14,18], we use the notations L to denote lattices defined by $\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m | \ \mathbf{A} \cdot \mathbf{e} = \mathbf{0}^n \bmod q\}$ or $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m | \ \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}$ w.l.o.g., where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Accordingly, use the notation $\mathcal{D}_{L,\sigma,\mathbf{c}}$ to denote the discrete Gaussian distributions of the support L, center $\mathbf{c} \in \mathbb{R}^m$ and parameter $\sigma > 0$, which is defined by $\mathcal{D}_{L,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(L)}$ for each $\mathbf{x} \in L$ where $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/\sigma^2)$ is the Gaussian function over $\mathbb{R}^m$. When $\mathbf{c} = \mathbf{0}$, we also write the Gaussian distributions as $\mathcal{D}_{L,\sigma}$ for short. The following fact ensures that the outputs of the discrete Gaussian distribution are always short.

Lemma 1. ([3]) *Given any $L \subseteq \mathbb{R}^n$ and $\sigma > 0$, $\Pr_{\mathbf{b} \hookleftarrow D_{L,\sigma}}[\|\mathbf{b}\| \leq \sqrt{n}\sigma] \geq 1 - 2^{-\Omega(n)}$.*

For appropriate parameters, the syndrome $\mathbf{u} = \mathbf{A} \cdot \mathbf{e}$ with $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{e} \in \mathbb{Z}_q^m$ is nearly uniform over $\mathbb{Z}_q^n$.

Lemma 2. ([18]) *Given positive integers n, q with q prime, let $m \geq 2n \log q$ and $s \geq \omega(\sqrt{\log m})$. Then for any $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, the distribution of the syndrome $\mathbf{u} = \mathbf{A} \cdot \mathbf{e} \bmod q$ is within negligible distance to the uniform distribution over $\mathbb{Z}_q^n$, where $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m,s}$.*

The computational lattice problems and associated hardness claims used in this work are stated as follows.

Definition 1 (SIS). *Given appropriate positive integers n, m, q, β, the* $\mathsf{SIS}_{n,m,q,\beta}$ *problem is defined as: for any* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$*, search a non-zero vector* $\mathbf{x} \in \mathbb{Z}^m$ *such that* $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ *and* $\|\boldsymbol{x}\| \leq \beta$*.*

By choosing appropriate parameters, the standard worst-case lattice problem SIVP_γ can be reduced to the average-case $\mathsf{SIS}_{n,m,q,\beta}$ problem. Such an example is followed by setting $m, \beta = \mathsf{poly}(n)$; $q \geq \sqrt{n}\beta$ and $\gamma = \widetilde{\mathcal{O}}(\sqrt{n}\beta)$ (e.g., [2,18,32]).

Definition 2 (LWE). *Given appropriate positive integers n, m, q, and a probability distribution on $\mathbb{Z}$ denoted as χ. For secret $\mathbf{s} \in \mathbb{Z}_q^n$, define $\mathbf{A}_{\mathbf{s},\chi}$ as the distribution generated by sampling $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and $e \hookleftarrow \chi$, and returning* ($\mathbf{a}$, $\mathbf{a}^{\mathrm{T}} \cdot\ \mathbf{s}+e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*. The goal of* $\mathsf{LWE}_{n,q,\chi}$ *is to distinguish m samples from $\mathbf{A}_{\mathbf{s},\chi}$ 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 $B \geq \sqrt{n}\omega(\log n)$, 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., [10,38,39]).

2.2 LNWX Lattice-Based Accumulators

The LNWX accumulator [31] is an updatable variant opposed to the static counterpart [26], and we will use it in our construction to achieve dynamic group users enrollments and membership revocations. The accumulator is built on a family of hash functions $\mathcal{H} = \{h_{\mathbf{A}} | \mathbf{A} \in \mathbb{Z}_q^{n\times m}\}$ with $\mathbf{A} = \big[\mathbf{A}_0 | \mathbf{A}_1\big] \in \mathbb{Z}_q^{n\times m}$ which hash $(\mathbf{u}_0, \mathbf{u}_1) \in (\{0,1\}^{nk})^2$ into $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathsf{bin}\big(\mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 \bmod q\big) \in \{0,1\}^{nk}$. Its security is ensured by the hardness of the SIS problem.

Informally, as in [4,12,36], the accumulator is defined by the algorithms (TSetup, TAcc, $\mathsf{TWitness}$, $\mathsf{TVerify}$, $\mathsf{TUpdate}$). Namely, for a Merkle-tree with $N = 2^\ell$ leaves, algorithm TSetup takes a random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ to form a hash function $h_{\mathbf{A}}$; Algorithm TAcc accumulates all values $R = \{\mathbf{d}_0, ..., \mathbf{d}_{N-1}\}$ of each length nk on leaves into the root $\mathbf{u}$ via the recursive computations shown as $\mathbf{u}_{b_1,...,b_i} = h_{\mathbf{A}}(\mathbf{u}_{b_1,...,b_i,0}, \mathbf{u}_{b_1,...,b_i,1})$ for any node at depth $i \in [\ell]$ and $\mathbf{u} = h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1)$, where $(b_1, ..., b_i) \in \{0,1\}^i$; Algorithm $\mathsf{TWitness}$ returns $\perp$ if $\mathbf{d} \notin R$, otherwise computes the witness $w = \big((j_1, ..., j_\ell), (\mathbf{u}_{j_1,...,j_{\ell-1},\bar{j_\ell}}, ..., \mathbf{u}_{j_1,\bar{j_2}}, \mathbf{u}_{\bar{j_1}})\big) \in \{0,1\}^\ell \times \big(\{0,1\}^{nk}\big)^\ell$ demonstrating that $\mathbf{d} = \mathbf{d}_j \in R$ for some $j \in [0, N-1]$ with $\mathsf{bin}(j) = (j_1, ..., j_\ell)$, where $\bar{b}$ denotes the bit $1 - b$ for a chosen bit b; Then, given a witness $w = \big((j_1, ..., j_\ell), (\mathbf{w}_\ell, ..., \mathbf{w}_1)\big) \in \{0,1\}^\ell \times \big(\{0,1\}^{nk}\big)^\ell$, and set $\mathbf{v}_\ell = \mathbf{d}$, algorithm $\mathsf{TVerify}$ computes the path $\mathbf{v}_{\ell-1}, ..., \mathbf{v}_0 \in \{0,1\}^{nk}$ via the recursive formula $\mathbf{v}_i = \bar{j}_{i+1} \cdot h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}) + j_{i+1} \cdot h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1})$ for any $j \in [0, N-1]$ and $i \in [\ell - 1]$ with initial setting $\mathbf{u} = \mathbf{v}_0$; Finally, when a value at position j is replaced by $\mathbf{p}$, algorithm $\mathsf{TUpdate}(\mathsf{bin}(j), \mathbf{p})$ efficiently updates the accumulator by simply updating the hash values of nodes on path from the specific leaf to the root, then the algorithm $\mathsf{TWitness}$ outputs the updated paths and maintains other values unchanged.

2.3 GPV Dual Encryption

The GPV encryption presented in [18] features the public-key anonymity and is efficient because of being free of lattice trapdoors. We now recall a variant that would be used in our construction. Choose positive integers n and $q \geq 2$ and set $k = \lfloor \log q \rfloor$ and $m = 2nk$. Select a random public matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Given a Gaussian parameter σ, a Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$ and an error distribution χ^m, one samples a short matrix $\mathbf{E}$ from $D^m_{\mathbb{Z}^m,\sigma}$ as the secret key sk, and computes a corresponding public matrix $\mathbf{U} = \mathbf{A} \cdot \mathbf{E} \in \mathbb{Z}_q^{n \times m}$ as the public key pk. To encrypt a message $\mathbf{m} \in \{0,1\}^m$, one samples a random vector $\mathbf{s} \hookleftarrow U(\{0,1\}^n)$ and two random vectors $\mathbf{x}, \mathbf{y} \hookleftarrow \chi^m$ to compute the ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ as: $\mathbf{c}_1 = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{x}, \mathbf{c}_2 = \mathbf{U}^\top \cdot \mathbf{s} + \mathbf{y} + \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil$. When the decryptor wants to recover the message $\mathbf{m}$, he uses the preserved key $\mathsf{sk} = \mathbf{E}$ to compute $\lfloor (\mathbf{c}_2 - \mathbf{E}^\top \cdot \mathbf{c}_1)/\frac{q}{2} \rceil$.

2.4 Zero-Knowledge Argument of Knowledge

A zero-knowledge argument system of knowledge (ZKAoK) is a two-party interactive protocol, where a prover $\mathcal{P}$ triggers a proof to convince the verifier $\mathcal{V}$ that he knows a witness of the specific statement while not revealing any additional information. More formally, given an NP relation defined by a set of statements-witnesses $R = \{(y, w)\} \in \{0,1\}^* \times \{0,1\}^*$, the associated ZKAoK is defined via an interactive game $\langle \mathcal{P}, \mathcal{V} \rangle$ with completeness δ_c and soundness error δ_s as:

- Completeness. For any given $(y, w) \in R$, $\Pr[\langle \mathcal{P}(y, w), \mathcal{V}(y) \rangle \neq 1] \leq \delta_c$.
- Soundness. Given any $(y, w) \notin R, \forall$ PPT $\widehat{\mathcal{P}}$: $\Pr[\langle \widehat{\mathcal{P}}(y, w), \mathcal{V}(y) \rangle = 1] \leq \delta_s$.

In the lattice setting, the Stern-like argument system [41] is a generic framework with statistical ZK property and soundness 2/3, and has been widely applied in the constructions of advanced cryptographic schemes [23,25,26,30]. Its key idea is to use "decomposition-extension-permutation" techniques to transform the targeted NP relations into those suitable for the framework, which in general increases double to four times communication cost and makes the system quite inefficient in practice together with soundness 2/3. In this work, we use a currently presented framework referred as Yang et al.'s argument system [43] which uses novel techniques to capture the computational ZK property and an inverse polynomial soundness. Let us recall it below.

The Abstraction of the Argument System. The desired ZKAoK system provided in Sect. 4 is covered within the following abstraction:

$$R = \{(\mathbf{M}, \mathbf{y}), (\mathbf{x}) : \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \wedge \mathbf{x} \in \mathsf{cond}\}, \tag{1}$$

where $\mathbf{M}, \mathbf{y}$ are the public matrix and vector, respectively, and the vector $\mathbf{x}$ is the secret witness, additionally cond represents the set of conditions that $\mathbf{x}$ should satisfy, which covers all possible constraints such as short vectors, quadratic relations. Actually, the set cond is always equally represented by a set $\mathcal{M} = \{(h, i, j)\}$ consisting of index tuples of $\mathbf{x}$ that satisfy the relation $\mathbf{x}[h] = \mathbf{x}[i] \cdot \mathbf{x}[j]$.

3 Model and Security Requirements of Fully Dynamic Group Encryption

In this section, by introducing a time factor and a group updating algorithm into the KTY model [21], also taking less oracles than that of [35], we provide the formalized model and security definitions of the fully dynamic group encryption (FDGE) primitive, which are appropriately upgraded and modified from the KTY model [21] that is only suitable for partially dynamic groups.

Like the KTY model [21], the FDGE also involves several parties: a group manager (GM) that managers a group of users, an opening authority (OA) that is empowered to revoke the anonymity of recipients should the misbehavior arise, and a set of prospective users as well as a sender producing well-formed ciphertexts for certified group members. In the forthcoming model, users join/leave the group under the permission of GM who can regularly edit and publish authentic group information info_τ at growing epoch τ, thereby anyone can learn the knowledge about changes of the group including, current/excluded group members. Additionally, by comparing two group information info_{τ_1} and info_{τ_2} under the convention that $\tau_1 < \tau_2$ if info_{τ_1} is published before info_{τ_2}, one can even identify revoked users at the recent epoch. The formalized fully dynamic group encryption is defined via the following tuple of algorithms:

- $\mathsf{SETUP}(\lambda)$: This algorithm consists of three procedures and generates group public key $\mathsf{gpk} = (\mathsf{pp}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}})$ as follows:
  - $\mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$: On input the security parameter λ, output public parameters pp.
  - $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{pp})$: Given pp, output the GM's key pair $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$.
  - $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{pp})$: Given pp, output a key pair $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the OA.

  An interaction occurs between the GM and the OA, successfully creating group public key gpk at its end, while the GM initializes the group information info and the registration table $\mathbf{reg}$.
- $\mathsf{UKGEN}(\mathsf{pp})$: On input pp, this algorithm produces a user key pair $(\mathsf{pk}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}})$.
- $\langle \mathsf{JOIN}(\mathsf{sk}_{\mathsf{U}}), \mathsf{ISSUE}(\mathsf{sk}_{\mathsf{GM}}) \rangle (\mathsf{info}_\tau, \mathsf{gpk}, \mathsf{pk}_{\mathsf{U}})$: This is an interaction run by the GM and a prospective user at epoch τ, whose successful completion enrolls a new group member with an identifier uid and makes the algorithm JOIN and algorithm ISSUE store group member secret key $\mathsf{sk}[\mathsf{uid}]$ and public key certificate $\mathsf{cert}_{\mathsf{pk}_{\mathsf{U}}}$ in the table $\mathbf{reg}$ with same index, respectively.
- $\mathsf{GUPDATE}(\mathsf{gpk}, \mathsf{sk}_{\mathsf{GM}}, \mathsf{info}_{\tau_{\mathrm{current}}}, \mathcal{S}, \mathbf{reg})$: Given $\mathsf{gpk}, \mathsf{sk}_{\mathsf{GM}}, \mathsf{info}_{\tau_{\mathrm{current}}}$, table $\mathbf{reg}$, a set $\mathcal{S}$ of active users to be removed, GM runs this algorithm to generate new group information $\mathsf{info}_{\tau_{\mathrm{current}}+1}$ and update the table $\mathbf{reg}$, while advancing the epoch and outputting $\perp$ if there is no change to the group.
- $\langle \mathcal{G}_r, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle (\mathsf{pp})$: Given pp, procedure $\mathsf{sample}_{\mathcal{R}}$ samples a statement-witness pair $(x, w) \in \mathcal{R}$ by using the key pair $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$ itself produced by procedure $\mathcal{G}_r$, where $\mathsf{sk}_{\mathcal{R}}$ may be empty in the most of real realizations.

- $\mathsf{ENC}(\mathsf{gpk}, \mathsf{pk}_\mathsf{U}, \mathsf{cert}_\mathsf{U}, \mathsf{info}_\tau, x, w, L)$: This algorithm is executed by sender to compute a group encryption Ψ on witness w with a label L under some public key pk_U. It returns $\perp$ if the target group user is inactive at epoch τ.
- $\mathsf{DEC}(\mathsf{sk}_\mathsf{U}, \Psi, L)$: The target receiver decrypts the ciphertext Ψ via this algorithm.
- $\mathsf{OPEN}(\mathsf{sk}_\mathsf{OA}, \mathsf{info}_\tau, \mathbf{reg}, \Psi, L)$: This algorithm is run by OA to return an identity uid of a group member who has secret information to decrypt the ciphertext together with a proof π attributing Ψ to user uid or to return $(\perp, \pi)$ if it fails to trace the receiver.
- $\langle \mathcal{P}(\mathsf{pk}_\mathsf{U}, \mathsf{cert}_\mathsf{U}, w, \mathit{coins}_\Psi), \mathcal{V}(\pi_\Psi)\rangle(\mathsf{gpk}, \mathsf{info}_\tau, x, \Psi, L)$: This is an interactive procedure run between sender and verifier which, given inputs, convinces verifier that the ciphertext Ψ is well-formed and is actually generated for one of active group members at epoch τ.

  For security requirements, as in [21], the FDGE scheme considers *correctness*, *message secrecy*, *anonymity* and *soundness*, whose definitions are given via corresponding experiments below, respectively.

Correctness asks that a ciphertext generated by a genuine sender is always decrypted successfully by algorithm DEC, and that procedure OPEN can always identify the receiver, as well as produces a proof that can be accepted by verifier.

Definition 3. *The correctness is satisfied if the following experiment returns* 1 *with negligible probability.*

Experiment $\mathbf{Exp}^{\mathsf{corr}}(\lambda)$

```
$\mathsf{pp} \leftarrow \mathsf{SETUP_{init}}(1^\lambda); (\mathsf{pk}_\mathcal{R}, \mathsf{sk}_\mathcal{R}) \leftarrow \mathcal{G}_\mathcal{R}(1^\lambda); (x, w) \leftarrow \mathsf{sample}_\mathcal{R}(\mathsf{pk}_\mathcal{R}, \mathsf{sk}_\mathcal{R});$
$(\mathsf{pk_{GM}}, \mathsf{sk_{GM}}) \leftarrow \mathsf{SETUP_{GM}}(\mathsf{pp}); (\mathsf{pk_{OA}}, \mathsf{sk_{OA}}) \leftarrow \mathsf{SETUP_{OA}}(\mathsf{pp});$
$\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert_{pk}} | \mathsf{uid}, \mathsf{pk}, \mathsf{cert_{pk}}, \mathsf{info}_\tau \rangle \leftarrow \langle \mathsf{J_{user}}, \mathsf{J_{GM}}(\mathsf{sk_{GM}}) \rangle(\mathsf{pk_{GM}}, \mathsf{info}_\tau);$
if $\mathsf{IsActive}(\mathsf{info}_\tau, \mathbf{reg}, \mathsf{uid}) = 0$, return 0.
$\Psi \leftarrow \mathsf{ENC}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}, \mathsf{cert_{pk}}, \mathsf{info}_\tau, w, L);$
$\pi_\Psi \leftarrow \mathcal{P}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}, \mathsf{cert_{pk}}, \mathsf{info}_\tau, x, w, \Psi, L, \mathit{coins}_\Psi).$
if $\big((w \neq \mathsf{DEC}(\mathsf{sk}, \Psi, L)) \vee (\mathsf{pk} \neq \mathsf{OPEN}(\mathsf{sk_{OA}}, \mathsf{info}_\tau, \mathbf{reg}, \Psi, L))$
    $\vee (\mathcal{V}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{info}_\tau, x, \Psi, L, \pi_\Psi) = 0)\big)$ then return 0 else return 1.
```

Message Secrecy demands that it is difficult for any PPT adversary to distinguish a ciphertext generated by a random plaintext from a one done by a specific relation pair, even if the adversary can corrupt all parties except the honest receiver via accessing to the following stateful and stateless oracles:

- $\mathsf{DEC}(\mathsf{sk}, \cdot)$: is a stateless decryption oracle with a restriction not to decrypt a ciphertext-label pair (Ψ, L) termed as $\mathsf{DEC}^{\neg\langle \Psi, L\rangle}$.
- $\mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, \tau, w, L)$: is a one-time oracle used for generating real-or-random challenge ciphertexts according to the choice of coin b at epoch τ. It returns $(\Psi, \mathit{coins}_\Psi)$ with $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}, \mathsf{cert_{pk}}, \mathsf{info}_\tau, w, L)$ if $b = 1$. Otherwise, return $(\Psi, \mathit{coins}_\Psi)$ with $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}, \mathsf{cert_{pk}}, \mathsf{info}_\tau, w', L)$ where w' is a uniformly random plaintext of length $\mathcal{O}(\lambda)$ sampled in the plaintext space, and coins_Ψ represents the random coins used to compute Ψ.

- $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk},\mathsf{cert_{pk}},\mathsf{info}_\tau,x,w,\Psi,L,coins_\Psi)$: is a stateful oracle that generates an actual proof π_Ψ or a simulated proof π'_Ψ for epoch τ by running the real prover $\mathcal{P}$ when $b=1$ and running the simulator $\mathcal{P}'$ else wise. It can be invoked a polynomial number times.

The usage of these oracles describes a experiment where the whole system is under the control of adversary except the member chosen as recipient. It shows the advantage of the adversary in mounting the attack against message secrecy.

Definition 4. *The message secrecy is achieved if, for any* PPT *adversary, the absolute difference of probability of outputting* 1 *between the following experiments* $\mathbf{Exp}_{\mathcal{A}}^{sec-1}(\lambda)$ *and* $\mathbf{Exp}_{\mathcal{A}}^{sec-0}(\lambda)$ *is negligible.*

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{sec}-b}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{SETUP_{init}}(1^\lambda)$; $(\mathsf{aux},\mathsf{pk_{GM}},\mathsf{pk_{OA}}) \leftarrow \mathcal{A}(\mathsf{pp})$;
$\langle \mathsf{pk},\mathsf{sk},\mathsf{cert_{pk}}|\mathsf{info}_\tau,\mathsf{aux}\rangle \leftarrow \langle \mathsf{J_{user}},\mathcal{A}(\mathsf{aux})\rangle(\mathsf{pk_{GM}},\mathsf{info}_\tau)$;
$(\mathsf{aux},x,w,L,\mathsf{pk}_{\mathcal{R}}) \leftarrow \mathcal{A}^{\mathsf{DEC}(\mathsf{sk},\cdot)}(\mathsf{aux})$; if $(x,w)\notin\mathcal{R}$ then return 0;
$b \hookleftarrow \{0,1\}$; $(\Psi,coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{ror}}(\lambda,\mathsf{pk},\tau,w,L)$;
$b' \leftarrow \mathcal{A}^{\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk},\mathsf{cert_{pk}},\mathsf{info}_\tau,x,w,\Psi,L,coins_\Psi),\mathsf{DEC}^{\neg\langle\Psi,L\rangle}(\mathsf{sk},\cdot)}(\mathsf{aux},\Psi)$;
Return b'.

Anonymity requires that it is infeasible for any PPT adversary to distinguish ciphertexts computed under two valid public keys of its choice, even if it controls the entire system except the OA and two well-behaved users via accessing the following oracles:

- $\mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk}_0,\mathsf{pk}_1,\mathsf{info}_\tau,w,L)$: is a challenge oracle that returns a pair $(\Psi,coins_\Psi)$ consisting of a ciphertext $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk}_b,\mathsf{cert}_{\mathsf{pk}_b},\mathsf{info}_\tau,w,L)$ and the coin tosses $coins_\Psi$ used for generating Ψ when a plaintext w and two possible public keys $\mathsf{pk}_0,\mathsf{pk}_1$ are given.
- $\mathsf{USER}(\mathsf{pk_{GM}},\tau)$: is a stateful oracle that simulates two instantiations of $\mathsf{J_{user}}$ via valid certificates $\{\mathsf{cert}_{\mathsf{pk}_b}\}_{b=0}^1$ supplied by adversarial GM in string keys at epoch τ, where honest outputs termed as $\{(\mathsf{pk}_b,\mathsf{sk}_b,\mathsf{cert}_{\mathsf{pk}_b})\}_{b=0}^1$ are stored.
- $\mathsf{OPEN}(\mathsf{sk_{OA}},\mathsf{info}_\tau,\mathbf{reg},\cdot)$: is a stateless oracle that executes opening operation on behalf of OA for the received ciphertext and reveals the identity of the receiver.

These above oracles can be used in a experiment that models the anonymity property, which reveals the advantage of adversary in this attack game.

Definition 5. *The* FDGE *scheme satisfies anonymity if, for any* PPT *adversary, the absolute difference of probability of outputting* 1 *between the following experiments* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-1}(\lambda)$ *and* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-0}(\lambda)$ *is negligible.*

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-b}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{SETUP_{init}}(1^\lambda)$; $(\mathsf{pk_{OA}},\mathsf{sk_{OA}}) \leftarrow \mathsf{SETUP_{OA}}(\mathsf{pp})$;
$(\mathsf{aux},\mathsf{pk_{GM}}) \leftarrow \mathcal{A}(\mathsf{pp},\mathsf{pk_{OA}})$; $\mathsf{aux} \leftarrow \mathcal{A}^{\mathsf{USER}(\mathsf{pk_{GM}},\tau),\mathsf{OPEN}(\mathsf{sk_{OA}},\mathsf{info}_\tau,\mathbf{reg},\cdot)}(\mathsf{aux})$;

```
if $\mathsf{keys} \neq (\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0}, \mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1}, \mathsf{info}_\tau)$ $(\mathsf{aux})$ then return 0;
$(\mathsf{aux}, x, w, L, \mathsf{pk}_\mathcal{R}) \leftarrow \mathcal{A}^{\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}_\tau, \tau, \cdot), \mathsf{DEC}(\mathsf{sk}_0, \cdot), \mathsf{DEC}(\mathsf{sk}_1, \cdot)}(\mathsf{aux})$;
if $(x, w) \notin \mathcal{R}$ return 0; $b \hookleftarrow \{0,1\}$; $(\Psi, coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1,$
$\mathsf{info}_\tau, w, L)$;
$b' \leftarrow \mathcal{A}^{\mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, \mathsf{info}_\tau, x, w, \Psi, L, coins_\Psi), \mathsf{OPEN}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}_\tau, \mathbf{reg}, \cdot),}$
        $^{\mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_0, \cdot), \mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_1, \cdot)}(\mathsf{aux}, \Psi)$. Return $b'$.
```

Soundness requires that it is infeasible for any PPT adversary to produce a convincing valid ciphertext that opens to unregistered group member or invalid public key, even if it can choose OA's key, and is given access to the REG oracle. In the following, database, $\mathcal{PK}$ and $\mathcal{C}$ are respectively used to represent the sets of registered public keys, valid keys and valid ciphertexts.

Definition 6. *An* FDGE *scheme is sound if, for any* PPT *adversary, the experiment below returns* 1 *with negligible probability.* Experiment $\mathbf{Exp}^{\mathsf{sound}}_\mathcal{A}(\lambda)$

```
$\mathsf{pp} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{pp})$;
$(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{pp})$;
$(\mathsf{pk}_\mathcal{R}, x, \Psi, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{aux}, \mathsf{info}_\tau) \leftarrow \mathcal{A}^{\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}}, \cdot)}(\mathsf{pp}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}, \mathsf{info}_\tau)$;
if $\mathcal{V}(\Psi, L, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{info}_\tau) = 0$ return 0;
$\mathsf{pk} \leftarrow \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}_\tau, \mathbf{reg}, \Psi, L)$; if$\big($ $(\mathsf{pk} \notin \mathsf{database}) \vee (\mathsf{pk} \notin \mathcal{PK}) \vee$
        $(\Psi \notin \mathcal{C}^{x, L, \mathsf{pk}_\mathcal{R}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}})\big)$ then return 1 else return 0.
```

To meet the above security requirement that pk must belong to the language of valid public keys, we use the Gaussian short vectors as shown in Sect. 5.1 to generate dense space for public keys, which simplifies our definitions.

4 The Underlying Zero-Knowledge Layer

In this section, we first introduce the needed decomposition techniques in Sect. 4.1. Then, we provide two generic and efficient zero-knowledge proofs for inequality relations of binary vectors (one is for *non-zero* binary vectors, and the other is for Hamming distance) that can work well in any lattice-based ZK framework and serve for our argument system. Finally, based on the techniques prepared in previous sections, we establish the argument system in Sect. 4.3 in the Yang et al.'s framework [43] recalled in Sect. 2.4. The argument system obtains great efficiency gains compared to that run in the Stern-type framework [41] since our system avoids using the "decomposition-extension-permutation" techniques (which at least increases the witness size double to four times) and also avoids repeating the protocol hundreds times (which incurs a drastic increase in communication cost) towards a negligible soundness as in [41].

4.1 Warm-Up: Decompositions

We briefly recall several decomposition techniques from [24,30] that would be used in constructing our argument system. We start with the integer decomposition function, i.e., for any non-negative integer i, let $\delta_i = \lceil \log(i+1) \rceil$, define

$\mathsf{bin}(i) = (i^{(1)}, ..., i^{(\delta_i)})^\top \in \{0,1\}^{\delta_i}$ and $\mathbf{g}_{\delta_i} = (1, 2, ..., 2^{\delta_i - 1})$, then it follows that $i = \sum_{j=1}^{\delta_i} 2^{j-1} \cdot i^{(j)} = \mathbf{g}_{\delta_i} \cdot \mathsf{bin}(i)$.

To decompose any integer $i \in [0, \beta]$ for a positive integer β, set $\delta_\beta \colon = \lceil \log_2(\beta + 1) \rceil$ and compute an integer sequence $\{\beta_1, ..., \beta_{\delta_\beta}\}$ via $\beta_j = \lfloor \frac{\beta + 2^{j-1}}{2^j} \rfloor, \forall j \in [1, \delta_\beta]$. Then, we have $i = \sum_{j=1}^{\delta_\beta} \beta_j \cdot i^{(j)} = \mathbf{g}'_{\delta_\beta} \cdot \mathsf{bin}'(\beta)$, where $\mathbf{g}'_{\delta_\beta} = (\beta_1, ..., \beta_{\delta_\beta})$ and $\mathsf{bin}'_\beta(i) = (i^{(1)}, ..., i^{(\delta_\beta)}) \in \{0,1\}^{\delta_\beta}$ which is a binary tuple computed in an interactive manner. This defines an integer decomposition function as $\mathsf{idec}_\beta(i) = (i^{(1)}, ..., i^{(\delta_\beta)})^\top \in \{0,1\}^{\delta_\beta}$ for any integer $i \in [0, \beta]$. Combining with $\mathbf{H}_{\mathfrak{m},\beta} = \mathbf{I}_{\mathfrak{m}} \otimes \mathbf{g}'_{\delta_\beta}$, we can similarly define decomposition functions for vectors and matrices (see, [25,26]):

- $\mathsf{vdec}_{\mathfrak{m},\beta} : [0,\beta]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_\beta}$ maps any β-bounded non-negative vector $\mathbf{v} = (v_1, ..., v_{\mathfrak{m}})^\top$ to $(\mathsf{idec}_\beta(v_1)^\top \| \ ... \ \| \mathsf{idec}_\beta(v_{\mathfrak{m}})^\top)^\top$ by applying $\mathsf{idec}_\beta(\cdot)$ to each entry of $\mathbf{v}$, which holds that $\mathbf{H}_{\mathfrak{m},\beta} \cdot \mathsf{vdec}_{\mathfrak{m},\beta}(\mathbf{v}) = \mathbf{v}$.
- $\mathsf{mdec}_{n,m,q} : \mathbb{Z}_q^{m \times n} \to \{0,1\}^{nm\delta_{q-1}}$ maps a matrix $\mathbf{X} = [\mathbf{x}_1 | \ ... \ | \mathbf{x}_n] \in \mathbb{Z}_q^{m \times n}$ to the size-$nm\delta_{q-1}$ binary vector $(\mathsf{vedc}_{m,q-1}(\mathbf{x}_1)^\top \| \ ... \ \| \mathsf{vedc}_{m,q-1}(\mathbf{x}_n)^\top)^\top$ by imposing $\mathsf{vdec}_{m,q-1}(\cdot)$ on the each column of $\mathbf{X}$ and concatenating the obtained binary vectors in the increasing order of the indexes of columns.

We note that, hereunder this section, when needing to decompose a bounded-β vector $\mathbf{v} \in [-\beta, \beta]^{\mathfrak{m}}$, we will first lift it to $\mathbf{v} + \boldsymbol{\beta} \in [0, 2\beta]^{\mathfrak{m}}$, then perform $\mathsf{vdec}_{\mathfrak{m},2\beta}(\cdot)$ on the transformed vector where $\boldsymbol{\beta} = (\beta, ..., \beta)$ consists of $\mathfrak{m}$'s β, with taking appropriate modifications for the involved matrices and vectors. This transformation-and-decomposition strategy will be quite useful for the construction of our ZK argument system.

4.2 Proving Inequality Relations for Binary Vectors

In this section, we first provide a ZK proof for demonstrating a binary vector $\mathbf{p}$ is *non-zero* (used to demonstrate a group user is activated) that can efficiently work well in any lattice-based ZK framework, on which we construct a ZK proof for the $\mathsf{Prohibitive}$ message filtering policy (used to demonstrate the validity of the encrypted witness) which is achieved over lattices at the first time and is generic and efficient. Startlingly, our proof methods can be extended to prove inequalities of general vectors, thus it is independent of interest.

Proving Binary Vectors $\mathbf{p} \neq \mathbf{0}$. Let n, q be positive integers with $n < q$ and $\mathbf{p} \in \{0,1\}^n$, our aim is to prove the secret $\mathbf{p} \neq \mathbf{0}$ in the Yang et al.'s framework [43]. Actually, this problem has been solved in the Stern-like framework [31] in spite of inefficiency and worse usability (i.e., it can not work in the Yang et al.'s framework [43]), where the system was established by appending $n-1$ "dummy" entries to extend the targeted vector $\mathbf{p} \in \{0,1\}^n$ to $\mathbf{p}' \in \{0,1\}^{2n-1}$ of Hamming weight n exactly and running the Stern-like protocol. To handle the task in the Yang et al.'s framework [43], one may find a possible solution in [27] where numerous lattice-based range arguments were developed to prove private integer relations such as $X \in [\alpha, \beta]$ for public integers $\alpha, \beta \geq 0$. But the techniques used there are invalid in proving that one knows at least a private X_j among a given set $\{X_1, ..., X_n\}$ each of which is bounded by $[\alpha_i, \beta_i]$ with $i \in [n]$ satisfies that $\alpha_j < X_j \leq \beta_j$, which essentially generalizes our problem when setting $\mathbf{p} = (X_1, ..., X_n)^\top$ and $\alpha_i = 0$ and $\beta_i = 1$ for all $i \in [n]$. We now develop new techniques to address this problem.

An important observation is that, the task to prove $\mathbf{p} \neq \mathbf{0}$ is equivalent to that proving that there is at least an entry of $\mathbf{p}$ is > 0. To end this, intuitively, it suffices to prove the $\mathbf{p}$'s Hamming weight is ≥ 1. In the following, we provide two efficient solutions, where the first is somewhat tedious, and then second is succinct and will be applied in the construction of our argument system.

Let $\mathbf{J}_n = (1, ..., 1)^\top \in \mathbb{Z}_q^n$ of which all entries are 1's. Suppose that the Hamming weight of binary vector $\mathbf{p}$ is ≥ 1, then we can establish our argument system by proving that one knows a complementary binary vector $\mathbf{q} \in \{0, 1\}^n$ with Hamming weight $\leq n - 1$ such that $\mathbf{p} + \mathbf{q} = \mathbf{J}_n \bmod q$. The inequality can be solved by decomposing $\mathbf{J}_n^\top \cdot \mathbf{q}$ via the vector $\mathbf{g}'_{\delta_\beta}$ with setting $\beta = n - 1$ as in Sect. 4.1. Then, it suffices for a prover to prove that he knows private vectors $\mathbf{p}, \mathbf{q} \in \{0, 1\}^n$ and $\mathbf{q}' \in \{0, 1\}^{\delta_{n-1}}$ such that the following conditions hold:

$$\begin{cases} \mathbf{p} + \mathbf{q} = \mathbf{J}_n \bmod q, \\ \mathbf{J}_n^\top \cdot \mathbf{q} = \mathbf{g}'_{\delta_{n-1}} \cdot \mathbf{q}' \bmod q. \end{cases} \tag{2}$$

Note that the above solution not only works well in the Yang et al.'s framework [43] but does well in the Stern-like framework [31], and is more efficient when used in the previous framework. In fact, to further achieve efficiency gains, we can directly go to prove the Hamming weight of $\mathbf{p}$ is ≥ 1, i.e., go to prove $\mathbf{J}_n^\top \cdot \mathbf{p} \geq 1$. Interestingly, we observe that the proof for this relation can be reduced to that one knows a secret non-negative integer $b \leq n - 1$ such that $\mathbf{J}_n^\top \cdot \mathbf{p} = 1 + b$. Combining with the decomposition techniques defined in Sect. 4.1, we equally write the relation as (assuming a private vector $\mathbf{q} \in \{0, 1\}^{\delta_{n-1}}$)

$$\mathbf{J}_n^\top \cdot \mathbf{p} - \mathbf{g}'_{\delta_{n-1}} \cdot \mathbf{q} = 1 \bmod q. \tag{3}$$

The last above solution is more efficient since it saves 50% size compared to the previous one, and both present solutions are generic and more efficient when working in [43] than that of [31]. Besides, our solutions can be readily extended to prove that one knows a private $\mathbf{x}$ having l_∞ or l_2 norm bounded by $[\alpha, \beta]$ with integers $\alpha, \beta \geq 0$.

Proving Bounded Hamming Distance. In general, there two commonly used message filtering policies termed as "Permisive" and "Prohibitive". Our task is to establish the argument system for the latter, and that for previous is trivial and is omitted in this work. Given positive integers $m \geq t \geq d$, and binary vectors $\mathbf{m} \in \{0, 1\}^m$ and $\mathbf{y}_i \in \{0, 1\}^t$ with $i \in [m - t + 1]$, we use $\mathbf{y}_i \sqsubset \mathbf{m}$ to mean that $\mathbf{y}_i$ is a substring of $\mathbf{m}$, i.e., there exist strings $\mathbf{x}_i, \mathbf{z}_i \in \{0, 1\}^{\leq m-t}$

such that $[\mathbf{x}_i^\top|\mathbf{y}_i^\top|\mathbf{z}_i^\top]^\top = \mathbf{m}$. Actually, the relation $\mathbf{y}_i \sqsubset \mathbf{m}$ is equivalent to the equality $\mathbf{B}_i \cdot \mathbf{m} = \mathbf{y}_i$ where $\mathbf{B}_i \in \mathbb{Z}_q^{t\times m}$ is a public matrix of the form $[\mathbf{0}|\mathbf{I}_t|\mathbf{0}]$. Now we define the message filtering policy "Prohibitive" used in this work:

$$R_{\mathsf{prohi}} = \{((\mathbf{s}_i)_{i=1}^e, \mathbf{m}) \in (\{0,1\}^t)^e \times \{0,1\}^m : d_H(\mathbf{s}_i, \mathbf{y}) \geq d, \forall i \in [e], \forall \mathbf{y} \sqsubset \mathbf{m})\}.$$

To build an argument system for the relation R_{prohi}, we begin with building a system for the simple relation $d_H(\mathbf{x}, \mathbf{y}) \geq d$ with $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ being public and secret. In the context of lattices, the proof is needed to be proceeded in mod q (involved with the dimension n for security, e.g., $q \geq \sqrt{n}$) instead of mod 2, which is always an open problem. Now we use a novel idea to address it. For any $x, y \in \{0,1\}$, we observe that $x \oplus y = x + y - 2x \cdot y$, which follows that $\mathbf{x} \oplus \mathbf{y} = \mathbf{x} + \mathbf{y} - 2(x_1 \cdot y_1, ..., x_n \cdot y_n)\top$ for binary vectors $\mathbf{x} = (x_1, ..., x_n)^\top$ and $\mathbf{y} = (y_1, ..., y_n)^\top$. Then, the task to prove $d_H(\mathbf{x}, \mathbf{y}) \geq d$ can be reduced to proving $\|\mathbf{x}+\mathbf{y}-2(x_1 \cdot y_1, ..., x_n \cdot y_n)^\top\|_1 \geq d$. By extending the proof method just developed above, in the setting of mod q, our task is reduced to proving that we hold a secret vector $\mathbf{z} \in \{0,1\}^{\delta_{n-d}}$ such that the following equation holds:

$$\mathbf{J}_n^\top \cdot (\mathbf{x} + \mathbf{y} - 2(x_1 \cdot y_1, ..., x_n \cdot y_n)^\top) - \mathbf{g}'_{\delta_{n-d}} \cdot \mathbf{z} = d \bmod q.$$

Based on the above result, for each $i \in [e], j \in [m-t+1]$, let $\mathbf{s}_i = (s_{i,1}, ..., s_{i,t})^\top$, $\mathbf{y}_j = \mathbf{B}_j \cdot \mathbf{m}$ with $\mathbf{y}_j = (y_{j,1}, ..., y_{j,t})$ and $\mathbf{B}_{j,1}^\top, ..., \mathbf{B}_{j,t}^\top$ be the row vectors of $\mathbf{B}_j$ (which essentially ensures that $y_{j,k} = \mathbf{B}_{j,k}^\top \cdot \mathbf{m}$). Then, the task to prove the relation R_{prohi} is equal to proving that one knows secret vectors $\mathbf{z}_{i,j} \in \{0,1\}^{\delta_{m-d}}$ such that $(\forall i \in [e], j \in [m-t+1])$:

$$\mathbf{J}_n^\top \cdot (\mathbf{s}_i + \mathbf{B}_j \cdot \mathbf{m} - 2(s_{i,1} \cdot \mathbf{B}_{j,1}^\top, ..., s_{i,t} \cdot \mathbf{B}_{j,t}^\top)^\top \cdot \mathbf{m}) - \mathbf{g}'_{\delta_{m-d}} \cdot \mathbf{z}_{i,j} = d \bmod q. \quad (4)$$

Then, let $\mathbf{B}_{i,j} = \mathbf{J}_n^\top \cdot (\mathbf{B}_j - 2(s_{i,1} \cdot \mathbf{B}_{j,1}^\top, ..., s_{i,t} \cdot \mathbf{B}_{j,t}^\top)^\top) \in \mathbb{Z}_q^{1\times m}$ and $d_{i,j} = d + \mathbf{J}_n^\top \cdot \mathbf{s}_i \in \mathbb{Z}_q$, which is followed by $\mathbf{B}_{[i]} = [\mathbf{B}_{i,1}^\top, ..., \mathbf{B}_{i,m-t+1}^\top]^\top \in \mathbb{Z}_q^{(m-t+1)\times m}$ and $\mathbf{B} = [\mathbf{B}_{[1]}^\top, ..., \mathbf{B}_{[e]}^\top]^\top \in \mathbb{Z}_q^{(m-t+1)e\times m}$. Accordingly, build $\mathbf{z}_{[i]} = [\mathbf{z}_{i,1}^\top, ..., \mathbf{z}_{i,m-t+1}^\top]^\top \in \mathbb{Z}_q^{(m-t+1)\delta_{m-d}}$, $\mathbf{z} = [\mathbf{z}_{[1]}^\top, ..., \mathbf{z}_{[e]}^\top]^\top \in \mathbb{Z}_q^{(m-t+1)e\delta_{m-d}}$, and $\mathbf{d}_{[i]} = [d_{i,1}, ..., d_{i,m-t+1}]^\top \in \mathbb{Z}_q^{m-t+1}$ and $\mathbf{d} = [\mathbf{d}_{[1]}^\top, ..., \mathbf{d}_{[e]}^\top]^\top \in \mathbb{Z}_q^{(m-t+1)e}$. Combining with the definition $\mathbf{I}_{\mathbf{g}'} = \mathbf{I}_{(m-t+1)e} \otimes \mathbf{g}'_{\delta_{m-d}}$, the relation R_{prohi} is equally written as:

$$[\mathbf{B}, \mathbf{I}_{\mathbf{g}'}] \cdot \begin{pmatrix} \mathbf{m} \\ \mathbf{z} \end{pmatrix} = \mathbf{d} \bmod q. \quad (5)$$

Run the above result in the Yang et al.'s framework [43], then the argument for bounded Hamming distance is established. It is seen that the above proof method is also generic and efficient.

4.3 The Underlying ZKAoK

We now state our argument system under the abstract framework provided in [43] as recalled in Sect. 2.4 for a wide of lattice relations to fulfill our intricate task.

Given the same settings of parameters as in Sect. 5.1, let $\mathsf{bin}(j) = (j_1, ..., j_\ell) \in \{0,1\}^\ell$, $\mathbf{j} = \mathsf{bin}(j)^\top$, $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ and $\mathbf{a}_{j,i} = \mathsf{mdec}_{n,m,q}(\mathbf{U}_{j,i}^\top)$ for each $i \in \{1,2\}$. As in [26,31], take the operator $\mathsf{ext}(\cdot,\cdot)$ to express $\mathsf{ext}(b, \mathbf{v}) = \begin{pmatrix} \bar{b} \cdot \mathbf{v} \\ b \cdot \mathbf{v} \end{pmatrix}$. Our protocol can be summarized as follows:

Public Input: Matrices $\mathbf{A}$, $\mathbf{G}$, $\mathbf{F}$, $\mathbf{B}$, $\mathbf{A}_{\mathsf{rec}}$, $\mathbf{A}_{\mathsf{oa}}$, $\mathbf{U}_{\mathsf{oa},1}$, $\mathbf{U}_{\mathsf{oa},2}$, $\mathbf{I}'_{\mathbf{g}}$, and vectors $\mathbf{u}_\tau$, $\mathbf{J}_{nk}$, $\mathbf{g}'_{\delta_{nk-1}}$, $\{\mathbf{c}^{(1)}_{\mathsf{rec},i}, \mathbf{c}^{(2)}_{\mathsf{rec},i}, \mathbf{c}^{(1)}_{\mathsf{oa},i}, \mathbf{c}^{(2)}_{\mathsf{oa},i}\}_{i \in \{1,2\}}$, $\mathbf{d}$.

Prover's Goal: Prove possession of the secret inputs in the following system

$$\begin{cases} \mathbf{j} = (j_1, ..., j_\ell)^\top, \big(\mathbf{p}_j, (\mathbf{w}^{(j)}_\ell, ..., \mathbf{w}^{(j)}_1)\big) \in (\{0,1\}^{nk})^{\ell+1} \text{ with } \mathbf{p}_j \neq \mathbf{0}, \\ \mathbf{q}_j \in \{0,1\}^{\delta_{nk-1}}, \mathbf{a}_{j,1}, \mathbf{a}_{j,2} \in \{0,1\}^{nmk}, \\ \mathbf{m} \in \{0,1\}^m, \mathbf{z} \in \{0,1\}^{(m-t+1)e\delta_{m-d}}, \\ i = 1,2 : \mathbf{s}_{\mathsf{rec},i}, \mathbf{s}_{\mathsf{oa},i} \in \{0,1\}^n, \\ \mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i}, \mathbf{x}_{\mathsf{oa},i} \in [-B, B]^m, \mathbf{y}_{\mathsf{oa},i} \in [-B, B]^\ell \end{cases} \tag{6}$$

such that the following system of modular linear equations holds:

$$\begin{cases} \mathbf{G} \cdot \mathbf{u}_\tau = \mathbf{A} \cdot \mathsf{ext}(j_1, \mathbf{v}^{(j)}_1) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_1, \mathbf{w}^{(j)}_1) \bmod q, \mathbf{v}^{(j)}_\ell = \mathbf{p}_j, \\ i \in [1, \ell - 1] : \\ \mathbf{0} = \mathbf{A} \cdot \mathsf{ext}(j_{i+1}, \mathbf{v}^{(j)}_{i+1}) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_{i+1}, \mathbf{w}^{(j)}_{i+1}) + (-\mathbf{G}) \cdot \mathbf{v}^{(j)}_i \bmod q, \\ 1 = \mathbf{J}_{nk}^\top \cdot \mathbf{p}_j + (-\mathbf{g}'_{\delta_{nk-1}}) \cdot \mathbf{q}_j \bmod q, \\ \mathbf{0} = \mathbf{G} \cdot \mathbf{p}_j + (-\mathbf{F}) \cdot (\mathbf{a}_{j,1}^\top || \mathbf{a}_{j,2}^\top)^\top \bmod q, \\ r = \{1,2\} : \mathbf{c}^{(1)}_{\mathsf{rec},r} = \mathbf{A}_{\mathsf{rec}}^\top \cdot \mathbf{s}_{\mathsf{rec},r} + \mathbf{x}_{\mathsf{rec},r} \bmod q, \\ \mathbf{c}^{(2)}_{\mathsf{rec},r} = \mathbf{U}_{j,r}^\top \cdot \mathbf{s}_{\mathsf{rec},r} + \mathbf{y}_{\mathsf{rec},r} + \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil \bmod q, \\ \mathbf{d} = [\mathbf{B}, \mathbf{I}'_{\mathbf{g}}] \cdot [\mathbf{m}^\top, \mathbf{z}^\top]^\top \bmod q, \\ \mathbf{c}^{(1)}_{\mathsf{oa},r} = \mathbf{A}_{\mathsf{oa}}^\top \cdot \mathbf{s}_{\mathsf{oa},r} + \mathbf{x}_{\mathsf{oa},r} \bmod q, \\ \mathbf{c}^{(2)}_{\mathsf{oa},r} = \mathbf{U}_{\mathsf{oa},r}^\top \cdot \mathbf{s}_{\mathsf{oa},r} + \mathbf{y}_{\mathsf{oa},r} + \mathbf{j} \cdot \lfloor \frac{q}{2} \rceil \bmod q, \end{cases} \tag{7}$$

To proceed the proof, we first build two argument systems Π_1 suitable for accumulator values problem and plain encryption, and Π_2 suitable for encryption with hidden matrices, respectively, then establish the final system Π_{GE} which covers all the above involved relations. The concrete steps are made as follows:

Build System Π_1. This system covers $(\ell + 6)$ equations consisting of the first $(\ell + 2)$ and the last four ones from the above equation system (7). Our task is to construct a ZKAoK system for the following relation:

$$R_1 = \{(\mathbf{M}_1, \mathbf{y}_1), (\mathbf{x}_1) : \mathbf{M}_1 \cdot \mathbf{x}_1 = \mathbf{y}_1 \wedge \mathbf{x}_1 \in \mathsf{cond}_1\}. \tag{8}$$

In the above, the matrix $\mathbf{M}_1$ consists of the involved public matrices and vectors $\{\mathbf{A}, \mathbf{G}, \mathbf{J}_{nk}, \mathbf{g}'_{\delta_{nk-1}}, \mathbf{F}, \mathbf{A}_{\mathsf{oa}}, \mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2}\}$ by an appropriate arrangement, and vectors $\mathbf{x}_1$ and $\mathbf{y}_1$ are similarly made by private inputs $\{\mathbf{j}, \{\mathbf{v}_i\}_i, \{\mathbf{w}_i\}_i, \mathbf{p}_j, \mathbf{q}_j, \mathbf{q}'_j, \{\mathbf{s}_{\mathsf{oa},i}\}_i, \{\mathbf{x}_{\mathsf{oa},i}\}_i, \{\mathbf{y}_{\mathsf{oa},i}\}_i\}$ and public vectors $\{\mathbf{G} \cdot \mathbf{u}_\tau, \mathbf{J}_{nk}, \{\mathbf{c}^{(1)}_{\mathsf{oa},i},$

$\mathbf{c}^{(2)}_{\mathsf{oa},i}\}_i\}$, and the cond_1 is the set of conditions that the private inputs should meet given in system (6). We now describe the constructions of desired variables.

We achieve our goal by a sequence of steps. Let $\mathbf{b}_1, \mathbf{b}_2$ be constant vectors, respectively, of the form $\mathbf{b}_1 = (B, ..., B)^\top \in \mathbb{Z}_q^m$ and $\mathbf{b}_2 = (B, ..., B)^\top \in \mathbb{Z}_q^\ell$. Then, conduct the following.

1. Transform the inputs bounded by some positive integer to ones with non-negative entries. Concretely, for each $i \in \{1, 2\}$, set $\mathbf{x}'_{\mathsf{oa},i} = \mathbf{x}_{\mathsf{oa},i} + \mathbf{b}_1 \in [0, 2B]^m$, and $\mathbf{y}'_{\mathsf{oa},i} = \mathbf{y}_{\mathsf{oa},i} + \mathbf{b}_2 \in [0, 2B]^\ell$.
2. Decompose the above newly transformed vectors $\mathbf{x}'_{\mathsf{oa},i}, \mathbf{y}'_{\mathsf{oa},i}$. For each $i \in \{1, 2\}$, apply the operator $\mathsf{vdec}(\cdot)$ defined in Sect. 4.1 to the above targeted vectors to produce binary vectors $\mathbf{x}''_{\mathsf{oa},i}, \mathbf{y}''_{\mathsf{oa},i}$ of size $m\delta_{2B}$ and $\ell\delta_{2B}$, respectively, such that $\mathbf{x}'_{\mathsf{oa},i} = \mathbf{H}_{m,2B} \cdot \mathbf{x}''_{\mathsf{oa},i}$ and $\mathbf{y}'_{\mathsf{oa},i} = \mathbf{H}_{\ell,2B} \cdot \mathbf{y}''_{\mathsf{oa},i}$.
3. Modify the involved public vectors accordingly. For each $i \in \{1, 2\}$, set $\mathbf{c}^{(1)'}_{\mathsf{oa},i} = \mathbf{c}^{(1)}_{\mathsf{oa},i} + \mathbf{b}_1$ and $\mathbf{c}^{(2)'}_{\mathsf{oa},i} = \mathbf{c}^{(2)}_{\mathsf{oa},i} + \mathbf{b}_2$.
4. Rewrite the first ℓ equations. For each $i \in [1, \ell]$, by $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ and the operator $\mathsf{ext}(\cdot, \cdot)$, we have $\mathbf{A} \cdot \mathsf{ext}(j_i, \mathbf{v}_i^{(j)}) + \mathbf{A} \cdot \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^{(j)}) = \mathbf{A}_1 \cdot \mathbf{v}_i + (\mathbf{A}_2 - \mathbf{A}_1) \cdot j_i \mathbf{v}_i + \mathbf{A}_2 \cdot \mathbf{w}_i + (\mathbf{A}_1 - \mathbf{A}_2) \cdot j_i \mathbf{w}_i$. Let $\mathbf{A}_{(1,2)} = [\mathbf{A}_1|\mathbf{A}_2 - \mathbf{A}_1|\mathbf{A}_2|\mathbf{A}_1 - \mathbf{A}_2]$, $\mathbf{A}_{[1,2]} = [-\mathbf{G}|\mathbf{0}_3|\mathbf{A}_1|\mathbf{A}_2 - \mathbf{A}_1|\mathbf{A}_2|\mathbf{A}_1 - \mathbf{A}_2]$ (where $\mathbf{0}_3$ means a block of form $[\mathbf{0}|\mathbf{0}|\mathbf{0}] \in (\mathbb{Z}_q^{n \times m})^3$) and $\mathbf{u}' = [(\mathbf{G} \cdot \mathbf{u}_\tau)^\top|\mathbf{0}^\top]^\top$, set a matrix $\mathbf{A}_{[1,\ell]} = \begin{pmatrix} \mathbf{A}_{(1,2)} & & \\ & \mathbf{A}_{[1,2]} & \\ & & \ddots \end{pmatrix}$ consisting of a $\mathbf{A}_{(1,2)}$ and $(\ell - 1)$'s $\mathbf{A}_{[1,2]}$ such that, for each $i \in [2, \ell - 1]$, the component $-\mathbf{G}$ from the i-th block $\mathbf{A}_{[1,2]}$ and the component $\mathbf{A}_1$ from the last block $\mathbf{A}_{(1,2)}$ or from the last $\mathbf{A}_{[1,2]}$ are in the same column. Accordingly, for each $i \in [1, \ell]$, we set $\mathbf{x}_{i,\mathbf{v}_i,\mathbf{w}_i} = [\mathbf{v}_i^\top|(j_i\mathbf{v}_i)^\top|\mathbf{w}_i^\top|(j_i\mathbf{w}_i)^\top]^\top$, and further set $\mathbf{x}_{\ell,\mathbf{v},\mathbf{w}} = [\mathbf{x}^\top_{1,\mathbf{v}_1,\mathbf{w}_1}|\cdots|\mathbf{x}^\top_{\ell,\mathbf{v}_\ell,\mathbf{w}_\ell}]^\top$, which gives that $\mathbf{u}' = \mathbf{A}_{[1,\ell]} \cdot \mathbf{x}_{\ell,\mathbf{v},\mathbf{w}}$.

After the above treatments, the targeted system is equally changed as:

$$\begin{cases} \mathbf{u}' = \mathbf{A}_{[1,\ell]} \cdot \mathbf{x}_{\ell,\mathbf{v},\mathbf{w}}, \\ 1 = \mathbf{J}_{nk}^\top \cdot \mathbf{p}_j + (-\mathbf{g}'_{\delta_{nk-1}}) \cdot \mathbf{q}_j, \\ \mathbf{0} = \mathbf{G} \cdot \mathbf{p}_j + (-\mathbf{F}) \cdot [\mathbf{a}_{j,1}^\top|\mathbf{a}_{j,2}^\top]^\top \bmod q, \\ i \in \{1, 2\} : \mathbf{c}^{(1)'}_{\mathsf{oa},i} = \mathbf{A}_{\mathsf{oa}} \cdot \mathbf{s}_{\mathsf{oa},i} + \mathbf{H}_{m,2B} \cdot \mathbf{x}''_{\mathsf{oa},i} \bmod q, \\ \mathbf{c}^{(2)'}_{\mathsf{oa},i} = \mathbf{U}_{\mathsf{oa},i}^\top \cdot \mathbf{s}_{\mathsf{oa},i} + \mathbf{H}_{\ell,2B} \cdot \mathbf{y}''_{\mathsf{oa},i} + \mathbf{j} \cdot \lfloor \frac{q}{2} \rfloor \bmod q. \end{cases} \tag{9}$$

Basing on the above preparations, we obtain the desired variables as follows:

1. Build the public matrix $\mathbf{M}_1$ and the public vector $\mathbf{y}_1$. Set $\mathbf{A}' := \mathbf{A}_{[1,\ell-1]}, \mathbf{A}'_1 := \mathbf{A}_2 - \mathbf{A}_1, \mathbf{A}'_2 := \mathbf{A}_1 - \mathbf{A}_2, \mathbf{I}'_\ell := \lfloor \frac{q}{2} \rfloor \cdot \mathbf{I}_\ell$, $\mathbf{g}' := -\mathbf{g}'_{\delta_{nk-1}}$, $\mathbf{F}' := -\mathbf{F}$, $\mathbf{G}' := -\mathbf{G}$ and $\mathbf{H}'_k := \mathbf{H}_{k,2B}$ with $k \in \{\ell, m\}$. Use the matrices in (9) to construct the desired matrix $\mathbf{M}_1$ and vector $\mathbf{y}_1$ as (here we abuse notation and use $[\mathbf{A}'^\top|\mathbf{G}'^\top]^\top$ to represent that the matrix $\mathbf{G}'$ and the component $\mathbf{A}_1$ from the last row of $\mathbf{A}'$ are in the same column)

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}' & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}' & \mathbf{0}_3 & \mathbf{A}_1 & \mathbf{A}_1' & \mathbf{A}_2 & \mathbf{A}_2' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{nk}^\top & \mathbf{g}' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\mathsf{oa}} & \mathbf{H}_m' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_\ell' & \mathbf{0} & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{\mathsf{oa},1}^\top & \mathbf{0} & \mathbf{H}_\ell' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\mathsf{oa}} & \mathbf{H}_m' & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_\ell' & \mathbf{0} & \mathbf{0}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{\mathsf{oa},2}^\top & \mathbf{0} & \mathbf{H}_\ell' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_3 & \mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}' \end{pmatrix}, \begin{pmatrix} \mathbf{u}' \\ 1 \\ \mathbf{c}_{\mathsf{oa},1}^{(1)'} \\ \mathbf{c}_{\mathsf{oa},1}^{(2)'} \\ \mathbf{c}_{\mathsf{oa},2}^{(1)'} \\ \mathbf{c}_{\mathsf{oa},2}^{(2)'} \\ \mathbf{0} \end{pmatrix}.$$

2. Build the private input $\mathbf{x}_1$. Arrange the modified private inputs shown in the system (9), establish the desired private vector $\mathbf{x}_1 = (\mathbf{j}^\top, \mathbf{x}_{\ell,\mathbf{v},\mathbf{w}}^\top, \mathbf{q}_j^\top, \mathbf{s}_{\mathsf{oa},1}^\top, \mathbf{x}_{\mathsf{oa},1}^{''\top}, \mathbf{y}_{\mathsf{oa},1}^{''\top}, \mathbf{s}_{\mathsf{oa},2}^\top, \mathbf{x}_{\mathsf{oa},2}^{''\top}, \mathbf{y}_{\mathsf{oa},2}^{''\top}, \mathbf{a}_{j,1}^\top, \mathbf{a}_{j,2}^\top)^\top$ with size n_1, where $n_1 = \ell + 2n + \delta_{nk-1} + 2(m+\ell)\delta_{2B} + 4\ell nk + 2nmk$.
3. Build the set of conditions cond_1. Let $\mathcal{M}_1$ be the set of triple indexes (h, i, l) of $\mathbf{x}_1$ with $h, i, l \in [n_1]$ such that $\mathbf{x}_1[h] = \mathbf{x}_1[i] \cdot \mathbf{x}_1[l]$. The set $\mathcal{M}_1$ is equivalent to the set cond_1. We now state the structure of $\mathcal{M}_1$:
   a. Observe that all entries of $\mathbf{x}_1$ are binary, we note that the choices $(h, i, l) = (i, i, i)_{i \in [n_1]}$ are in the set $\mathcal{M}_1$.
   b. Now consider the corresponding choices of $\mathcal{M}_1$ for $j_i \mathbf{v}_i, j_i \mathbf{w}_i$ for all $i \in [\ell]$: for $j_i \mathbf{v}_i$, the choices consist of $(h, i, l) = (\ell + (4i' - 3)nk + l', i', \ell + (4i' - 4)nk + l')_{i' \in [\ell], l' \in [nk]}$. Whereas, for $j_i \mathbf{w}_i$, the desired indexes are given by $(h, i, l) = (\ell + (4i' - 1)nk + l', i', \ell + (4i' - 2)nk + l')_{i' \in [\ell], l' \in [nk]}$.

This constructs the argument system Π_1 for the relation R_1, and by running the protocol in Sect. 4.3, the desired argument system is obtained.

Build System Π_2. This system covers the remaining five equations from the system (7). Our task is to construct a similar ZKAoK system for the following relation:

$$R_2 = \{(\mathbf{M}_2, \mathbf{y}_2), (\mathbf{x}_2) : \mathbf{M}_2 \cdot \mathbf{x}_2 = \mathbf{y}_2 \wedge \mathbf{x}_2 \in \mathsf{cond}_2\}. \tag{10}$$

As in the above system Π_1, the involved variables are respectively defined. We take similar strategies to proceed the present task.

1. For each $i \in \{1, 2\}$, transform the private inputs $\mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i}$ to ones that only have non-negative entries. Concretely, set $\mathbf{x}_{\mathsf{rec},i}' = \mathbf{x}_{\mathsf{rec},i} + \mathbf{b}_1, \mathbf{y}_{\mathsf{rec},i}' = \mathbf{y}_{\mathsf{rec},i} + \mathbf{b}_1, \in [0, 2B]^m$.
2. Decompose the above newly generated vectors. For each $i \in \{1, 2\}$, impose the function $\mathsf{vdec}(\cdot)$ on these vectors, respectively, to yield size-$m\delta_{2B}$ binary vectors $\mathbf{x}_{\mathsf{rec},i}''$ and $\mathbf{y}_{\mathsf{rec},i}''$ such that $\mathbf{x}_{\mathsf{rec},i}' = \mathbf{H}_{m,2B} \cdot \mathbf{x}_{\mathsf{rec},i}'', \mathbf{y}_{\mathsf{rec},i}' = \mathbf{H}_{m,2B} \cdot \mathbf{y}_{\mathsf{rec},i}''$.

3. Change the corresponding public matrices and vectors. Consider the decomposition of $\mathbf{U}_{j,i}^{\top} \cdot \mathbf{s}_{\mathsf{rec},i}$ with $i = 1, 2$. Let $\mathbf{U}_{j,i}^{\top} = [\mathbf{u}_{j,i}^{(1)\top} | ... | \mathbf{u}_{j,i}^{(n)\top}] \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{s}_{\mathsf{rec},i} = (s_{\mathsf{rec},i}^{(1)}, ..., s_{\mathsf{rec},i}^{(n)})^{\top} \in \{0,1\}^n$. In light of operators $\mathsf{vdec}(\cdot)$ and $\mathsf{mdec}(\cdot)$, we have $\mathbf{U}_{j,i}^{\top} \cdot \mathbf{s}_{\mathsf{rec},i} = \Sigma_{t=1}^{n} \mathbf{u}_{j,i}^{(t)\top} \cdot s_{\mathsf{rec},i}^{(t)} = \Sigma_{t=1}^{n} \mathbf{H}_{m,q-1} \cdot \mathbf{a}_{j,i}^{(t)} \cdot s_{\mathsf{rec},i}^{(t)} = \mathbf{H}_{m,q-1} \cdot \mathbf{s}_{\mathsf{rec},i,mk}^{\top} \cdot \mathbf{a}_{j,i}$, where $\mathbf{a}_{j,i}^{(t)} \in \{0,1\}^{mk}$ is the binary decomposition of the vector $\mathbf{u}_{j,i}^{(t)\top}$ and $\mathbf{s}_{\mathsf{rec},i,mk} = (\overbrace{s_{\mathsf{rec},i}^{(1)}, ..., s_{\mathsf{rec},i}^{(1)}}^{mk's \text{ times}}, ..., \overbrace{s_{\mathsf{rec},i}^{(n)}, ..., s_{\mathsf{rec},i}^{(n)}}^{mk's \text{ times}})^{\top}$. Additionally, for all $i = 1, 2$, set vectors as: $\mathbf{c}_{\mathsf{rec},i}^{(1)'} = \mathbf{c}_{\mathsf{rec},i}^{(1)} + \mathbf{b}_1$ and $\mathbf{c}_{\mathsf{rec},i}^{(2)'} = \mathbf{c}_{\mathsf{rec},i}^{(2)} + \mathbf{b}_1$.

After making the above treatments, the targeted system is equally changed as:

$$\begin{cases} i \in \{1,2\} : \mathbf{c}_{\mathsf{rec},i}^{(1)'} = \mathbf{A}_{\mathsf{rec}}^{\top} \cdot \mathbf{s}_{\mathsf{rec},i} + \mathbf{H}_{m,2B} \cdot \mathbf{x}_{\mathsf{rec},i}'' \bmod q, \\ \mathbf{c}_{\mathsf{rec},i}^{(2)'} = \mathbf{H}_{m,q-1} \cdot \mathbf{s}_{\mathsf{rec},i,mk}^{\top} \cdot \mathbf{a}_{j,i} + \mathbf{H}_{m,2B} \cdot \mathbf{y}_{\mathsf{rec},i}'' + \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil \bmod q, \\ \mathbf{d} = [\mathbf{B}, \mathbf{I}_{\mathbf{g}}'] \cdot [\mathbf{m}^{\top}, \mathbf{z}^{\top}]^{\top} \bmod q, \end{cases} \tag{11}$$

This proceeds the following constructions of variables.

1. For simplicity, let $\mathbf{H}_m'' = \mathbf{H}_{m,q-1}$ and $\mathbf{I}_m' = \lfloor \frac{q}{2} \rfloor \mathbf{I}_m$. Similar to what in system $\mathit{\Pi}_1$, build the public matrix $\mathbf{M}_2$ and the public vector $\mathbf{y}_2$ as

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}_{\mathsf{rec}}^{\top} & \mathbf{0} & \mathbf{H}_m' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_m'' & \mathbf{0} & \mathbf{H}_m' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_m' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\mathsf{rec}}^{\top} & \mathbf{0} & \mathbf{H}_m' & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_m'' & \mathbf{0} & \mathbf{H}_m' & \mathbf{I}_m' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{I}_{\mathbf{g}}' \end{pmatrix} \text{ and } \begin{pmatrix} \mathbf{c}_{\mathsf{rec},1}^{(1)'} \\ \mathbf{c}_{\mathsf{rec},1}^{(2)'} \\ \mathbf{c}_{\mathsf{rec},2}^{(1)'} \\ \mathbf{c}_{\mathsf{rec},2}^{(2)'} \\ \mathbf{d} \end{pmatrix}.$$

2. Build the private input $\mathbf{x}_2$. According to the public variables $\mathbf{M}_2$ and $\mathbf{y}_2$ above, we build the private vector $\mathbf{x}_2 = (\mathbf{a}_{j,1}^{\top}, \mathbf{a}_{j,2}^{\top}, \mathbf{s}_{\mathsf{rec},1}^{\top}, (\mathbf{s}_{\mathsf{rec},1,mk}^{\top} \cdot \mathbf{a}_{j,1})^{\top}, \mathbf{x}_{\mathsf{rec},1}^{''\top}, \mathbf{y}_{\mathsf{rec},1}^{''\top}, \mathbf{s}_{\mathsf{rec},2}^{\top}, (\mathbf{s}_{\mathsf{rec},2,mk}^{\top} \cdot \mathbf{a}_{j,2})^{\top}, \mathbf{x}_{\mathsf{rec},2}^{''\top}, \mathbf{y}_{\mathsf{rec},2}^{''\top}, \mathbf{m}^{\top}, \mathbf{z}^{\top})^{\top}$ which has size $n_2 = m + 2n + 2m(k + nk + 2\delta_{2B}) + (m - t + 1)e\delta_{m-d}$.
3. Consider the set of conditions cond_2. Similarly, let $\mathcal{M}_2$ be the set of triple indexes (h, i, l) of $\mathbf{x}_2$ with $h, i, l \in [n_2]$ such that $\mathbf{x}_2[h] = \mathbf{x}_2[i] \cdot \mathbf{x}_2[l]$. It can be seen that the defined set of indexes is equal to the original set cond_2. Now we present the structure of set $\mathcal{M}_2$:
   a. Observe that all components of $\mathbf{x}_2$ are binary vectors, which gives that such indexes $(h, i, l) = (i, i, i)$ with $i \in [n_2]$ are in cond_2.
   b. In addition, the hidden matrix constraint in the original system is equally to the conditions $\mathbf{s}_{\mathsf{rec},i,mk}^{\top} \mathbf{a}_{j,i} = s_{\mathsf{rec},i,mk}^{\top} \cdot \mathbf{a}_{j,i}$ for each $i \in \{1,2\}$ as in system (11). This allows us to compute another choice of indexes $(h, i, l) = (2nmk + n + (i' - 1)mk + l', 2nmk + i', l')_{i' \in [n], l' \in [mk]} \cup (3nmk + 2n + 2m\delta_{2B} + (i' - 1)mk + l', 3nmk + n + 2m\delta_{2B} + i', nmk + l')_{i' \in [n], l' \in [mk]}$.

This completes the task of constructing argument system $\mathit{\Pi}_2$ by running the protocol given in Sect. 4.3.

Build System $\mathit{\Pi}_{\mathsf{GE}}$. The final system is the desired one which covers the system $\mathit{\Pi}_1$ and the system $\mathit{\Pi}_2$ simultaneously, whose definition is shown as follows:

$$R_{\mathsf{GE}} = \{(\mathbf{M}, \mathbf{y}), (\mathbf{x}) : \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \wedge \mathbf{x} \in \mathsf{cond}\}. \tag{12}$$

To build the system, we write $\mathbf{M}_1 = [\mathbf{M}_{1,1} | \mathbf{M}_{1,2}]$ and $\mathbf{M}_2 = [\mathbf{M}_{2,1} | \mathbf{M}_{2,2}]$, then build $\mathbf{M} = \begin{pmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{2,1} & \mathbf{M}_{2,2} \end{pmatrix}$, where the blocks $\mathbf{M}_{1,2}$ and $\mathbf{M}_{2,1}$ respectively represent the last column and the first column of $\mathbf{M}_1$ and $\mathbf{M}_2$. Accordingly, we build $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 / \{[\mathbf{a}_{j,1}^\top | \mathbf{a}_{j,2}^\top]^\top\} \end{pmatrix}$, $\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$ and $\mathsf{cond} = \mathsf{cond}_1 \cap \mathsf{cond}_2$, then a system that is suitable for the framework established in [43] is obtained. Now the family $\mathcal{M}$ of triples corresponding to the set cond is somewhat modified, i.e., $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2'$, where $\mathcal{M}_2' = \{(h, i, l)\} = \{(i, i, i)\}_{i \in [n_1+1, n_1+n_2-2nmk]} \cup (n_1 + n + (i'-1)mk + l', n_1 + i', n_1 - 2nmk + l')_{i' \in [n], l' \in [mk]} \cup (n_1 + nmk + 2n + 2m\delta_{2B} + (i'-1)mk + l', n_1 + nmk + n + 2m\delta_{2B} + i', n_1 - nmk + l')_{i' \in [n], l' \in [mk]}$. Then, the prover runs an interactive protocol with the verifier as shown in [43], and the desired ZKAoK system is established.

5 Our Fully Dynamic Lattice-Based Group Encryption

This section describes how to make use of the LNWX accumulator [31], GPV dual encryption [18] and the ZKAoK system built in Sect. 4 to construct our fully dynamic lattice-based group encryption in a relatively simple manner. In our design, this scheme first achieves the "$\mathsf{Prohibitive}$" message filtering policy in the lattice setting and is free of lattice trapdoors throughout the design, resulting into great efficiency gains. All of these efforts yield a much more practical group encryption, also secure against the potential quantum attacks. We now briefly interpret the overview of our techniques.

Our inspiration begins with a main observation that, by using an updatable accumulator [31], one can directly upgrade the static group signature scheme [26] to one offering full dynamicity [31] at a reasonable cost, where the GM creates and revokes group membership via altering the hash value $\mathbf{p}$ of user's public key (*non-zero* for activated users and $\mathbf{0}$ otherwise). Following the idea, combining with the GPV dual encryption [18], we consider: For a group of $N = 2^\ell$ members, given $\mathbf{A}_{\mathsf{rec}} \in \mathbb{Z}_q^{n \times m}$, users sample two random short matrices $\mathbf{E}_{j,1}, \mathbf{E}_{j,2} \in \mathbb{Z}_q^{m \times m}$ from a given Gaussian distribution to generate nearly uniform $\mathbf{U}_{j,i} = \mathbf{A} \cdot \mathbf{E}_{j,i} \in \mathbb{Z}_q^{n \times m}$ with $i \in \{1, 2\}$, resulting secret/public key pairs $(\mathsf{sk}_j, \mathsf{pk}_j) = (\mathbf{E}_{j,1}, (\mathbf{U}_{j,1}, \mathbf{U}_{j,2}))$ with hash values $\mathbf{p}_j = \mathsf{bin}(\mathbf{F} \cdot [\mathbf{a}_{j,1}^\top | \mathbf{a}_{j,2}^\top]^\top) \in \{0,1\}^{nk}$ where $\mathbf{a}_{j,i} = \mathsf{mdec}_{n,m,k}(\mathbf{U}_{j,i}^\top) \in \{0,1\}^{nmk}$. Then, the manager builds an efficiently updatable tree on top of values $\mathbf{p}_0, \cdots, \mathbf{p}_{N-1}$ and publishes the tree root $\mathbf{u}$ as well as the witness for the fact $\mathbf{p}_j$ was accumulated in $\mathbf{u}$. Particularly, the GM conducts: (i)-For an invalid user who has not joined the group or has been excluded from the group, set the j-th leaf value $\mathbf{p}_j$ as $\mathbf{0}$; (ii)-For a valid user who joins the group and has not left the group, set the corresponding value as $\mathbf{p}_j$, the hash value of the public key pk_j; (iii)-With these rules, the GM can build an efficiently updatable tree with comparative complexity $\mathcal{O}(\log N)$, for which he only needs to alter the values at specific leaves and along their paths to the root rather than to reconstruct the whole tree when group information changes. These executions guarantee that all active users (with $\mathbf{p} \neq \mathbf{0}$) in the given epoch can be accumulated into the dynamic root while no any inactive user cannot, which effectively separates active users who can receive the valid ciphertexts from those who cannot in any growing epoch.

When moving to the stage of generating a group encryption, the sender fetches the public key $(\mathbf{U}_{j,1}, \mathbf{U}_{j,2})$ and the associated membership witness $w^{(j)}$ of the target group member from group information, then samples a witness in light of the given $\mathsf{Prohibitive}$ message filtering policy and computes the ciphertext (we apply the Naor-Yung transformation technique [34] for CCA-2 security) and an associated proof which shows that the ciphertext is well-formed and $\mathbf{p}_j \neq \mathbf{0}$. In order for the proof to work in the Yang et al.'s ZK framework [43], we use the proof techniques we just provided in Sect. 4.2 and then resort to the argument system built in Sect. 4.3.

We also note that the dynamicity described in [31] is de facto limited to once enrollment and once revocation. To realize stronger dynamicity that users are allowed to join or leave the group at will, some modifications on procedures $\langle \mathsf{JOIN}, \mathsf{ISSUE} \rangle$ and $\mathsf{GUPDATE}$ are needed. Concretely, we take some significant modifications for the procedures of user registering and user leaving, such that group users indeed obtain the expected dynamicity as long as their reasonable applications are accepted by the GM.

5.1 Description of the Scheme

As in [35], we assume that our scheme allows encrypting witness $\mathbf{m} \in \{0,1\}^m$ that meets both message filtering policies termed as $\mathsf{Permissive}$[1] and $\mathsf{Prohibitive}$ (shown in Sect. 4.2), which use constraints stronger than those used in [21,25]. For simplicity, we only take the latter policy in our scheme. Procedures of constructing the FDGE scheme are shown as follows.

- $\mathsf{SETUP}_{\text{init}}$ (1^λ): This algorithm conducts the following:
  - Set the possibly maximum number of group users as $N = 2^\ell = \mathsf{poly}(\lambda)$.
  - Select integer $n = \mathcal{O}(\lambda)$ and prime $q = \widetilde{\mathcal{O}}(n^2)$. Let $k = \lceil \log q \rceil$, $m = 2nk$.
  - Pick a discrete distribution χ over $\mathbb{Z}$ of the bound $B = \sqrt{n}\omega(\log n)$.
  - Select a Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$, and build a discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ with upper bound $\beta = \sigma \cdot \omega(\log n)$.
  - Take public parameters $\mathsf{pp}_{\mathsf{COM}}$ for the homomorphic commitment scheme like [5] which serves as a key building block in the construction of the interactive game $\langle \mathcal{P}, \mathcal{V} \rangle$.
  - Pick a random matrix $\mathbf{F} \hookleftarrow \mathbb{Z}_q^{n \times 2nmk}$ which hashes users' public keys from $\mathbb{Z}_q^{n \times 2m}$ to $\mathbb{Z}_q^n$.

[1] It is defined as $R_{\mathsf{permi}} = \{((\mathbf{s}_i)_{i=1}^e, \mathbf{m}) \in (\{0,1\}^t)^e \times \{0,1\}^m : \exists i \in [e] \text{s.t.} \mathbf{s}_i \sqsubset \mathbf{m}\}$.

  - Set a gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}_k$ with the definition given in Sect. 4.1. Pick matrices $\mathbf{A}_{\mathsf{rec}}, \mathbf{A}_{\mathsf{oa}} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ that will be used to generate public keys for group users and the opening authority, respectively.

  Output

$$\mathsf{pp}= \{N, \ell, \lambda, n, q, k, m, B, \chi, \sigma, \beta, \mathsf{pp}_{\mathsf{COM}}, \mathbf{F}, \mathbf{G}, \mathbf{A}_{\mathsf{rec}}, \mathbf{A}_{\mathsf{oa}}\}.$$

- $\mathsf{SETUP}_{\mathsf{GM}}$ (pp): This algorithm picks a random matrix $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2] \hookleftarrow \mathbb{Z}_q^{n\times m}$ consisting of two same-size matrices, and samples $\mathsf{sk}_{\mathsf{GM}} \hookleftarrow \{0,1\}^m$ and computes $\mathsf{pk}_{\mathsf{GM}} = \mathbf{A} \cdot \mathsf{sk}_{\mathsf{GM}}$, resulting a key pair $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$ for the GM. Here, we take $\mathsf{pk}_{\mathsf{GM}}$ as an identifier of the group and assume that only the GM (i.e., the party holding $\mathsf{sk}_{\mathsf{GM}}$) can edit and publish the group information.
- $\mathsf{SETUP}_{\mathsf{OA}}$ (pp): This procedure samples two short secret matrices $\mathbf{E}_{\mathsf{oa},i}$ with $i \in \{1,2\}$ from the distribution $D^{\ell}_{\mathbb{Z}^m,\sigma}$ to generate two corresponding matrices $\mathbf{U}_{\mathsf{oa},i} = \mathbf{A}_{\mathsf{oa}} \cdot \mathbf{E}_{\mathsf{oa},i} \in \mathbb{Z}_q^{n\times \ell}$, which forms the secret key $\mathsf{sk}_{\mathsf{OA}} = \mathbf{E}_{\mathsf{oa},1} \in \mathbb{Z}_q^{m\times \ell}$ and the public key $\mathsf{pk}_{\mathsf{OA}} = (\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2}) \in (\mathbb{Z}_q^{n\times \ell})^2$ for the OA.

  When GM receives $\mathsf{pk}_{\mathsf{OA}}$ sent from the OA, it executes the following:
  1. Build table **reg:** $=(\{\mathbf{reg}[j][i]\}_{j\in[0,N-1],i\in\{1,2\}})$ initialized as $\mathbf{reg}[j][1] = \mathbf{0}^{nk}$ and $\mathbf{reg}[j][2] = 0$. Note that the former records the user's registered public key, while the latter stores the epoch at which an execution of joining protocol is performed.
  2. Build a Merkle tree $\mathcal{T}$ on top of $\{\mathbf{reg}[j][1]\}_{j\in[0,N-1]}$ whose initial values are zero and then changed with users' public keys by the GM when one successfully joins the group or the group executes an updating operation.
  3. Set the counter of users $c := 0$.

  Then, GM outputs $\mathsf{gpk} = (\mathsf{pp}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}})$ and publicizes the initial group information $\mathsf{info} = \emptyset$, while $\mathcal{T}$ as well as c is kept by him self.
- $\mathsf{UKGEN}(\mathsf{pp})$: For each $j \in [0, N-1]$ and each $i \in \{1,2\}$, user U_j samples two secret matrices $\mathbf{E}_{j,i}$ from the Gaussian distribution $D^{m}_{\mathbb{Z}^m,\sigma}$ to generate two corresponding public matrices $\mathbf{U}_{j,i} = \mathbf{A}_{\mathsf{rec}} \cdot \mathbf{E}_{j,i} \in \mathbb{Z}_q^{n\times m}$, which forms the secret key $\mathsf{sk}_j = \mathbf{E}_{j,1} \in \mathbb{Z}_q^{m\times m}$ and the public key $\mathsf{pk}_j = (\mathbf{U}_{j,1}, \mathbf{U}_{j,2}) \in (\mathbb{Z}_q^{n\times m})^2$. Then, the user computes a hash value $\mathbf{p}_j = \mathsf{bin}(\mathbf{F} \cdot (\mathbf{a}_{j,1}^{\top}||\mathbf{a}_{j,2}^{\top})^{\top}) \in \{0,1\}^{nk}$ with $\mathbf{a}_{j,i} = \mathsf{mdec}_{n,m,q}(\mathbf{U}_{j,i}^{\top}) \in \{0,1\}^{nmk}$ for each $i \in \{1,2\}$. We note that all honestly generated pk_j's are non-zero and pairwise distinct, since the probability that users take zero-matrix $\mathbf{U}_{j,i}$ or same matrix (i.e., $\mathbf{U}_{j,i} = \mathbf{U}_{j',i'}$ for some $j \neq j'$ or $i \neq i'$), or finds a collision for hash function $\mathbf{F}$ is negligible (due to the assumed hardness of the SIS problem).
- $\langle\mathsf{JOIN}(\mathsf{sk}); \mathsf{ISSUE}(\mathsf{sk}_{\mathsf{GM}})\rangle(\mathsf{gpk}, \mathsf{pk}, \mathsf{info}_{\tau})$: Let $\mathcal{S}_0$ be a set of indexes i of which associated public keys of group users are zero, with the initialization $\{\mathbf{reg}[j][1]\}$. When a user holding key pair $(\mathsf{pk}, \mathsf{sk})$ with binary hash $\mathbf{p}$ wants to join the group at the epoch τ, he sends $\mathbf{p}$ to the GM who proceeds the following procedures with him after the request is accepted:
  1. GM picks a random $j \in \mathcal{S}_0$ and sets a member identifier $\mathsf{bin}(j) \in \{0,1\}^{\ell}$ for the user, and executes the following:
     - Update $\mathcal{T}$ by running procedure $\mathsf{TUpdate}_{\mathbf{A}}(\mathsf{bin}(j), \mathbf{p}_j)$.

- Register the user to table $\mathbf{reg}$ as $\mathbf{reg}[j][1] := \mathbf{p}_j$.
- Update the set $\mathcal{S}_0 := \mathcal{S}_0 - \{j\}$, increase the counter $c := c + 1$.

2. When specific enrollment requests at a same epoch are ending, basing on the above updated results (note that the update process is essentially like that of running algorithm $\mathsf{TAcc}_{\mathbf{A}}(\cdot)$ on $\mathbf{reg}[\cdot][1] = \{\mathbf{reg}[j][1]\}_j$ for the generation of root value $\mathbf{u}$, thus same results are led), the GM runs algorithm $\mathsf{TWitness}_{\mathbf{A}}(\mathbf{reg}[\cdot][1], \mathbf{p}_j)$ to output a witness

$$w^{(j)} = \big((j_1, ..., j_\ell) \in \{0,1\}^\ell, (\mathbf{w}_\ell^{(j)}, ..., \mathbf{w}_1^{(1)}) \in (\{0,1\})^\ell\big)$$

to the fact that $\mathbf{p}_j$ is accumulated in $\mathbf{u}$.
3. User checks the validity of $w^{(j)}$ by algorithm $\mathsf{TVerify}_{\mathbf{A}}(\mathbf{u}, \mathbf{p}_j, w^{(j)})$ and outputs $\perp$ if it is unaccepted. Otherwise, set $\mathsf{wit}_j = (\mathbf{u}, w^{(j)})$ as the witness of pk_j being accumulated into the root $\mathbf{u}$, which plays the similar role to a certificate of public key issued by the GM.

- $\mathsf{GUPDATE}(\mathsf{gpk}, \mathsf{sk}_{\mathsf{GM}}, \mathsf{info}_{\tau_{\text{current}}}, \mathcal{S}, \mathbf{reg})$: GM updates the group information while advancing the epoch by running this algorithm as follows.
  1. Let $\mathcal{S}$ be a set of verified public keys of group users to be removed. If $\mathcal{S} = \emptyset$, go to Step 2. Otherwise, let $\mathcal{S} = \{\mathbf{reg}[j_i][1]\}_{i=1}^r$ for some $r \in [1, N]$ and $j_i \in [0, N-1]$ for all $i \in [r]$, then GM runs $\mathsf{TUpdate}_{\mathbf{A}}(\mathsf{bin}(j_i), \mathbf{0}^{nk})$ to update the tree $\mathcal{T}$, followed by $\mathcal{S}_0 := \mathcal{S}_0 \bigcup \mathcal{S}$.
  2. By construction, each zero-value leaf in $\mathcal{T}$ corresponds to an inactive user, i.e., one that is revoked or has not yet got membership. This means that only active users capable of decrypting well-formed ciphertexts generated in the new epoch τ_{new} will have *non-zero* hash values of public keys $\{\mathbf{p}_j\}_j$, that are accumulated in the root $\mathbf{u}_{\tau_{\text{new}}}$ of the updated tree.
  For each j, let $w^{(j)} \in \{0,1\}^\ell \times (\{0,1\}^{nk})^\ell$ be the witness showing that $\mathbf{p}_j$ is accumulated in $\mathbf{u}_{\tau_{\text{new}}}$. GM publishes the updated group information:

$$\mathsf{info}_{\tau_{\text{new}}} = \big(\mathbf{u}_{\tau_{\text{new}}}, \{w^{(j)}\}_j\big).$$

  As described below, in order to verify ciphertexts bound to epoch τ, the verifier only needs to download the first component $\mathbf{u}_\tau$ of size $\widetilde{\mathcal{O}}(\lambda)$ bits. Meanwhile, to compute a well-formed ciphertext, it is sufficient for sender to download the witness of size $\widetilde{\mathcal{O}}(\ell\lambda)$ of some active user.
- $\langle \mathcal{G}_r, \mathsf{sample}_{\mathcal{R}} \rangle$: Algorithm $\mathcal{G}_r$ outputs parameters (t, e) for the $\mathsf{Prohibitive}$ policy to form ($\mathsf{pk}_{\mathcal{R}}$, $\mathsf{sk}_{\mathcal{R}}$)= $((t, e), \varepsilon)$. Then algorithm $\mathsf{sample}_{\mathcal{R}}$ takes $\mathsf{pk}_{\mathcal{R}}$ as input, and returns a set $\{\mathbf{s}_1, ..., \mathbf{s}_e\} \in (\{0,1\}^t)^e$ and a witness $\mathbf{m} \in \{0,1\}^m$ such that they hold for the relation R_{prohi} (i.e., meet the Eq. (5)).
- $\mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_j, \mathsf{wit}_j, \mathsf{info}_\tau, \{\mathbf{s}_i\}_{i=1}^e, \mathbf{m}, L)$: To encrypt the sampled witness $\mathbf{m}$ with the group information info_τ at epoch τ, sender first checks whether a witness associated with $\mathsf{bin}(j)$ is contained in info_τ. If it is not this case, return $\perp$. Otherwise, the sender downloads $\mathbf{u}_\tau$ and some witness $(\mathsf{bin}(j), (\mathbf{w}_\ell, ..., \mathbf{w}_1))$ from info_τ, then parses $\mathsf{pk}_{\mathsf{OA}}$ as $(\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2})$ and wit_j as $(\mathbf{u}_\tau, w^{(j)})$ for some $j \in [0, N-1]$, and proceeds as follows.

1. Encrypt the witness $\mathbf{m} \in \{0,1\}^m$ under U_j's public key $\mathsf{pk}_j \in (\mathbb{Z}_q^{n \times m})^2$. For each $i \in \{1,2\}$, randomly take a tuple $(\mathbf{s}_{\mathsf{rec},i}, \mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i}) \in U(\{0,1\}^n) \times (\chi^m)^2$ to form the private parameter set $\mathsf{rand}_{\mathsf{rec}} = (\mathbf{s}_{\mathsf{rec},i}, \mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i})_{i\in\{1,2\}}$. Compute the corresponding ciphertext $\mathbf{c}_{\mathsf{rec},i} = (\mathbf{c}^{(1)}_{\mathsf{rec},i}, \mathbf{c}^{(2)}_{\mathsf{rec},i}) \in (\mathbb{Z}_q^m)^2$ as
$$\mathbf{c}^{(1)}_{\mathsf{rec},i} = \mathbf{A}^{\top}_{\mathsf{rec}} \cdot \mathbf{s}_{\mathsf{rec},i} + \mathbf{x}_{\mathsf{rec},i} \bmod q, \mathbf{c}^{(2)}_{\mathsf{rec},i} = \mathbf{U}^{\top}_{j,i} \cdot \mathbf{s}_{\mathsf{rec},i} + \mathbf{y}_{\mathsf{rec},i} + \mathbf{m} \cdot \lfloor \frac{q}{2} \rceil, \quad (13)$$
which follows the ciphertext $\mathbf{c}_{\mathsf{rec}} = (\mathbf{c}_{\mathsf{rec},1}, \mathbf{c}_{\mathsf{rec},2}) \in (\mathbb{Z}_q^m \times \mathbb{Z}_q^m)^2$.
2. Encrypt the user identifier $\mathbf{j} \in \{0,1\}^\ell$ of user U_j by taking similar operations as above. First take a random tuple $(\mathbf{s}_{\mathsf{oa},i}, \mathbf{x}_{\mathsf{oa},i}, \mathbf{y}_{\mathsf{oa},i}) \in U(\{0,1\}^n) \times \chi^m \times \chi^\ell$ for each $i \in \{1,2\}$, which forms the private randomness set $\mathsf{rand}_{\mathsf{oa}} = (\mathbf{s}_{\mathsf{oa},i}, \mathbf{x}_{\mathsf{oa},i}, \mathbf{y}_{\mathsf{oa},i})_i$. Compute the corresponding ciphertext $\mathbf{c}_{\mathsf{oa},i} = (\mathbf{c}^{(1)}_{\mathsf{oa},i}, \mathbf{c}^{(2)}_{\mathsf{oa},i}) \in (\mathbb{Z}_q^m \times \mathbb{Z}_q^\ell)$ as
$$\mathbf{c}^{(1)}_{\mathsf{oa},i} = \mathbf{A}^{\top}_{\mathsf{oa}} \cdot \mathbf{s}_{\mathsf{oa},i} + \mathbf{x}_{\mathsf{oa},i} \bmod q, \mathbf{c}^{(2)}_{\mathsf{oa},i} = \mathbf{U}^{\top}_{\mathsf{oa},i} \cdot \mathbf{s}_{\mathsf{oa},i} + \mathbf{y}_{\mathsf{oa},i} + \mathbf{j} \cdot \lfloor \frac{q}{2} \rceil, \quad (14)$$
which follows the identity ciphertext $\mathbf{c}_{\mathsf{oa}} = (\mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2}) \in (\mathbb{Z}_q^m \times \mathbb{Z}_q^\ell)^2$.

Finally, put the above ciphertexts together, we obtain the ciphertext $\Psi = \big(\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}\big)$ and the state information $coins_\Psi = \big(\mathsf{rand}_{\mathsf{rec}}, \mathsf{rand}_{\mathsf{oa}}\big)$.

- $\mathsf{DEC}(\mathsf{sk}_j, \Psi, L)$: This algorithm takes the following steps to decrypt Ψ:
1. Parse the secret key sk_j as $\mathbf{E}_{j,1}$ and the ciphertext Ψ as $\big(\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}\big)$.
2. Use the secret key $\mathbf{E}_{j,1}$ to proceed the decryption of $\mathbf{c}_{\mathsf{rec}}$ as
$$\mathbf{m} = \left\lfloor \left(\mathbf{c}^{(2)}_{\mathsf{rec},1} - \mathbf{E}^{\top}_{j,1} \cdot \mathbf{c}^{(1)}_{\mathsf{rec},1}\right) / \left\lfloor \frac{q}{2} \right\rceil \right\rceil. \quad (15)$$
Then, output $\mathbf{m}$ if it satisfies the relation R_{prohi}. Otherwise, return $\perp$.

- $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \mathsf{info}_\tau, \mathbf{reg}, \Psi, L)$: This algorithm decrypts the ciphertext $\mathbf{c}_{\mathsf{oa}} = (\mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2})$ by proceeding the following steps:
1. Parse the secret key $\mathsf{sk}_{\mathsf{oa}}$ as $\mathbf{E}_{\mathsf{oa},1}$ and the ciphertext Ψ as $\big(\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}\big)$.
2. To reveal the targeted recipient, use $\mathbf{E}_{\mathsf{oa},1}$ to decrypt the $\mathbf{c}_{\mathsf{oa},1}$ as
$$\mathbf{j}' = \left\lfloor \left(\mathbf{c}^{(2)}_{\mathsf{oa},1} - \mathbf{E}^{\top}_{\mathsf{oa},1} \cdot \mathbf{c}^{(1)}_{\mathsf{oa},1}\right) / \left\lfloor \frac{q}{2} \right\rceil \right\rceil. \quad (16)$$
3. Check that whether the group information info_τ includes a witness containing $\mathbf{j}'$ or not, and return $\perp$ if it is not this case.
4. Let $j' \in [0, N-1]$ be the integer whose binary decomposition is $\mathbf{j}'$, if $\mathbf{reg}[j'][1] = \mathbf{0}^{nk}$ in table $\mathbf{reg}$, then return $\perp$.

- $\langle \mathcal{P}(\mathsf{pk}_j, \mathsf{wit}_j, \mathbf{m}, coins_\Psi), \mathcal{V}(\pi_\Psi)\rangle(\mathsf{gpk}, \mathsf{info}_\tau, \{\mathbf{s}_i\}_{i=1}^e, \Psi, L)$: Given the common inputs $\mathsf{gpk}, \mathsf{info}_\tau, \{\mathbf{s}_i\}_{i=1}^e, \Psi$ and L. The prover's secret inputs consist of a witness $\mathbf{m} \in \{0,1\}^m$, $\mathsf{pk}_j = (\mathbf{U}_{j,1}, \mathbf{U}_{j,2}) \in (\mathbb{Z}_q^{n\times m})^2$, certificate $\mathsf{wit}_j = (\mathbf{u}_\tau, w^{(j)})$ and random coins $coins_\Psi = \big(\mathbf{s}_{\mathsf{rec},i}, \mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i}; \mathbf{s}_{\mathsf{oa},i}, \mathbf{x}_{\mathsf{oa},i}, \mathbf{y}_{\mathsf{oa},i}\big)_{i\in\{1,2\}}$, while the verifier takes π_Ψ as its private input.

The prover constructs a zero-knowledge argument system π_Ψ to convince the verifier that the secret inputs he makes satisfy the following conditions (details of which are shown in Sect. 4):

- $\mathbf{G} \cdot \mathbf{p}_j = \mathbf{F} \cdot (\mathbf{a}_{j,1}^\top || \mathbf{a}_{j,2}^\top)^\top \bmod q$.
- $\mathsf{TVerify}_{\mathbf{A}}(\mathbf{u}, \mathbf{p}_j, w^{(j)}) = 1$ and $\mathbf{p}_j \neq \mathbf{0}$.
- Witness $\mathbf{m}$ satisfies the relation R_{prohi} defined in Sect. 4.2.
- For each $i \in \{0, 1\}$, vectors $\mathbf{s}_{\mathsf{rec},i}, \mathbf{s}_{\mathsf{oa},i}$ are of the form $\{0, 1\}$, and vectors $\mathbf{x}_{\mathsf{rec},i}, \mathbf{y}_{\mathsf{rec},i}, \mathbf{x}_{\mathsf{oa},i}, \mathbf{y}_{\mathsf{oa},i}$ have infinity B-bounded norm.
- Equations of (13) and (14) hold.

Correctness. The correctness of the proposed group encryption follows from correctly decrypting the GPV dual ciphertexts, which may cause some decryption errors. Indeed, during the decryption procedure of $\mathsf{DEC}(\mathsf{sk}_j, \Psi, L)$, we have:

$$\mathbf{c}_{\mathsf{rec},1}^{(2)} - \mathbf{E}_{j,1}^\top \cdot \mathbf{c}_{\mathsf{rec},1}^{(1)} = \mathbf{y}_{\mathsf{rec},1} - \mathbf{E}_{j,1}^\top \cdot \mathbf{x}_{\mathsf{rec},1} + \mathbf{m} \cdot \left\lfloor \frac{q}{2} \right\rfloor. \tag{17}$$

Note that $\|\mathbf{x}_{\mathsf{rec},1}\|_\infty$ and $\|\mathbf{y}_{\mathsf{rec},1}\|_\infty$ both have upper bound B, and $\|\mathbf{E}_{j,1}\|_\infty$ is bounded by β. Then $\|\mathbf{y}_{\mathsf{rec},1} - \mathbf{E}_{j,1}^\top \cdot \mathbf{x}_{\mathsf{rec},1}\|_\infty \leq B + m\beta B$ and is further bounded by $\widetilde{\mathcal{O}}(n^{1.5})$ which is smaller than $q/5 = \widetilde{\mathcal{O}}(n^2)$. As a result, the decryption algorithm returns $\mathbf{m}$ with overwhelming probability. This gives the correctness of $\mathsf{DEC}(\mathsf{sk}_j\ , \Psi, L)$. For $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \Psi, L)$, a similar analysis is proceeded and $\|\mathbf{y}_{\mathsf{oa},1} - \mathbf{E}_{\mathsf{oa},1}^\top \cdot \mathbf{x}_{\mathsf{oa},1}\|_\infty$ is also bounded by $\widetilde{\mathcal{O}}(n^{1.5})$.

Finally, we argue that if a sender honestly follows all the prescribed algorithms for the specific certified group user, valid witness-vectors to be used in the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ are able to be computed and the present proof is accepted by the verifier, thanks to the completeness of the argument system in Sect. 4.3.

5.2 Analysis of the Scheme

Security Analysis. We provide provable security analysis for our scheme under the SIS and LWE hardness assumptions via the classical reduction methods. These security results and associated proofs are shown in the following.

Theorem 1. *The anonymity is satisfied if the $\mathsf{LWE}_{n,q,\chi}$ assumption holds.*

Proof. We prove the anonymity using a sequence of indistinguishable games, where we begin with running the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-0}$ and end with the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-1}$ from Definition 5 to show that the advantage for the adversary succeeding in the last game is negligible. For simplicity, hereunder we take PPT algorithms $\mathcal{A}$ and $\mathcal{B}$ as the adversary and challenger, respectively, and denote by W_i the event that the adversary $\mathcal{A}$ returns $b' = 1$ in game i.

Game 1: This is the real experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{anon}-0}$ except that $\mathcal{B}$ retains $\mathbf{E}_{\mathsf{oa},2}$, which makes no any difference in the adversary's view since $\mathbf{E}_{\mathsf{oa},2}$ is not used in the following real experiment. Concretely, the challenger $\mathcal{B}$ publicizes the parameters pp containing $\mathbf{A}_{\mathsf{rec}}, \mathbf{A}_{\mathsf{oa}} \in \mathbb{Z}^{n \times m}, \mathbf{F} \in \mathbb{Z}_q^{n \times 2nmk}$ as a part, and sends the opening public key $\mathsf{pk}_{\mathsf{OA}} = (\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2}) \in (\mathbb{Z}_q^{n \times m})^2$ to $\mathcal{A}$ who certifies the honest group members on behalf of GM by invoking the USER oracle. Specially, after receiving two users' public keys $\mathsf{pk}_0 = (\mathbf{U}_{0,1}, \mathbf{U}_{0,2}) \in (\mathbb{Z}_q^{n \times m})^2$ and $\mathsf{pk}_1 =$

$(\mathbf{U}_{1,1}, \mathbf{U}_{1,2}) \in (\mathbb{Z}_q^{n\times m})^2$ of challenger's choice, $\mathcal{A}$ registers the keys in the table **reg** and conducts a number of queries w.r.t. opening and decryption algorithms, whose response is handled by $\mathcal{B}$ by using $\mathsf{sk_{OA}} = \mathbf{E}_{\mathsf{oa},1}$ and $\mathsf{sk}_0 = \mathbf{E}_{0,1}, \mathsf{sk}_1 = \mathbf{E}_{1,1}$. Then, the adversary moves to the challenge phase to provide a valid witness $\mathbf{m} \in \{0,1\}^m$ satisfying the $\mathsf{Prohibitive}$ for challenge. In return, the challenger takes the bit $b = 0$ and computes a group encryption $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ of the witness $\mathbf{m}$ under $\mathsf{pk}_b = (\mathbf{U}_{b,1}, \mathbf{U}_{b,2})$, and the user identity $\mathbf{j}_b = \mathbf{j}_0$ under $\mathsf{pk_{oa}} = (\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2})$ with $\mathbf{c}_{\mathsf{oa}} = (\mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2})$, which follows real proofs $\pi^*_{\Psi^*}$ of Ψ^* and queries of opening and decryption under the natural restrictions of the security definition. When $\mathcal{A}$ halts, it returns a bit $b' \in \{0,1\}$ and the challenger $\mathcal{B}$ returns 1 iff $b' = b$. Otherwise, $\mathcal{B}$ outputs 0 indicating that the adversary fails in this game, which gives the success probability $\Pr[W_1 = 1]$.

Game 2: This game is like Game 1 except one change in executing the ciphertext opening oracle $\mathsf{OPEN}(\mathsf{sk_{oa}}, .)$. Concretely, $\mathcal{B}$ uses $\mathbf{E}_{\mathsf{oa},2} \in \mathbb{Z}_q^{m\times \ell}$ instead of $\mathsf{sk_{oa}} = \mathbf{E}_{\mathsf{oa},1} \in \mathbb{Z}_q^{m\times \ell}$ to decrypt $\mathbf{c}_{\mathsf{oa}}$ among the ciphertext $\Psi = (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}})$. It can be seen that, in the $\mathcal{A}$'s view, this game is the same as Game 1 until the event F_1 that $\mathcal{A}$ queries the opening oracle $\mathsf{OPEN}(\mathsf{sk_{oa}}, .)$ for a ciphertext $\Psi = (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2})$ where $\mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2}$ encrypt two distinct ℓ-size identities. By the soundness of our argument presented in Sect. 4.3, $\Pr[W_2] - \Pr[W_1]$ is bounded by $\Pr[F_1]$ which itself is bounded by $\mathbf{Adv}^{\mathsf{sound}}_{\mathcal{B}}(\lambda)$.

Game 3: This game is identical to Game 2 except a modification in the generation of proofs $\pi^*_{\Psi^*}$. Instead of employing the real random coins $coins^*_{\Psi} = (\{\mathbf{s}^*_{\mathsf{rec},i}\}_i, \{\mathbf{x}^*_{\mathsf{rec},i}\}_i, \{\mathbf{y}^*_{\mathsf{rec},i}\}_i, \{\mathbf{s}^*_{\mathsf{oa},i}\}_i, \{\mathbf{x}^*_{\mathsf{oa},i}\}_i, \{\mathbf{y}^*_{\mathsf{oa},i}\}_i)$ used for Ψ^* to generate proofs, we employ the zero-knowledge simulator of argument system described in Sect. 4.3 once invoking $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}$ after the challenge phase (note that, given trusted public parameters, the computationally indistinguishable simulation is achieved via the techniques [17] without increasing the number of rounds). Here the computational ZK property ensures that, for any PPT adversary, the change is unnoticed: $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$.

Game 4: This game is same as Game 3 except that we modify the generation of $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ with $\mathbf{c}^*_{\mathsf{oa}} = (\mathbf{c}^*_{\mathsf{oa},1}, \mathbf{c}^*_{\mathsf{oa},2})$ by encrypting a random size-ℓ identity $\mathbf{j}_1$ as $\mathbf{c}^*_{\mathsf{oa},1}$, while still retaining $\mathbf{c}^*_{\mathsf{oa},2}$ for the encryption of the index $\mathbf{j}_0$ corresponding to user U_0. By the semantic security of GPV dual encryption [18] (assuming the hardness of LWE problem) for public key $\mathsf{pk_{oa}} = (\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2})$, this game is identical to Game 3, i.e., $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

Game 5: This game makes one change by switching back to the application of $\mathbf{E}_{\mathsf{oa},1} \in \mathbb{Z}_q^{m\times \ell}$ for the $\mathsf{OPEN}(\mathsf{sk_{oa}}, \cdot)$ queries with discarding $\mathbf{E}_{\mathsf{oa},2}$, and the modification is invariant to the adversary except the event F_2, where the queries to the DEC for a valid ciphertext Ψ containing $\mathbf{c}^*_{\mathsf{oa},1}, \mathbf{c}^*_{\mathsf{oa},2}$ encrypting distinct ℓ-size identities $\mathbf{j}_0$ and $\mathbf{j}_1$, 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 $|\Pr[W_5 = 1] - \Pr[W_4 = 1]| \leq \mathbf{Adv}^{\mathsf{sound}}_{\Pi_{\mathsf{GE}}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 6: Here, this experiment performs a modification to the Game 5 only by taking $\mathbf{c}_{\mathsf{oa},2}$ as the encryption of $\mathbf{j}_1$ for the challenge ciphertext $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$

with $\mathbf{c}^*_{\mathsf{oa}} = (\mathbf{c}^*_{\mathsf{oa},1}, \mathbf{c}^*_{\mathsf{oa},2})$. Note that this change is unnoticed to $\mathcal{A}$ due to the semantic security the encryption shares for public key $\mathbf{U}_{\mathsf{oa},2}$, and also for the application of $\mathbf{E}_{\mathsf{oa},1}$ to the OPEN, we have $|\Pr[W_6 = 1] - \Pr[W_5 = 1]| = \mathsf{negl}(\lambda)$.

Game 7: This experiment generates a real proof for ciphertext $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ instead of using simulated proof, which is the only modification different from Game 6. The computational zero-knowledgeness of the underlying ZKAoK system makes the difference between Game 6 and Game 7 negligible, i.e., $\Pr[W_6 = 1] \approx \Pr[W_7 = 1]$. This is actually the experiment $\mathbf{Exp}^{\mathsf{anon}-1}_{\mathcal{A}}(\lambda)$, which directly leads that $\Pr[W_7 = 1] = \mathbf{Exp}^{\mathsf{anon}-1}_{\mathcal{A}}(\lambda)$. By these above games, we have $|\mathbf{Exp}^{\mathsf{anon}-1}_{\mathcal{A}}(\lambda) - \mathbf{Exp}^{\mathsf{anon}-0}_{\mathcal{A}}(\lambda)| = \mathsf{negl}(\lambda)$. This proves the anonymity. □

Theorem 2. *The message secrecy is satisfied if the $\mathsf{LWE}_{n,q,\chi}$ assumption holds.*

Proof. In a similar manner to that used in proving Theorem 1, we complete the proof via a sequence of indistinguishable games in which the first one is exactly the experiment $\mathbf{Exp}^{\mathsf{sec}-1}_{\mathcal{A}}$ which generates a real ciphertext and an associated real proof while the last one is the experiment $\mathbf{Exp}^{\mathsf{sec}-0}_{\mathcal{A}}$ that outputs a random ciphertext and an associated simulated proof. For simplicity, we use $\mathcal{A}, \mathcal{B}$ to represent the adversary and challenger, respectively. In addition, we also denote by W_i the event that the adversary $\mathcal{A}$ returns $b' = 1$ in game i.

Game 1: This is the real experiment $\mathbf{Exp}^{\mathsf{sec}-1}_{\mathcal{A}}$ except that $\mathcal{B}$ retains $\mathbf{E}_{j,2}$, which makes no any difference in the adversary's view since $\mathbf{E}_{j,2}$ is not used in the following real experiment. Concretely, $\mathcal{A}$ is first fed with public parameters pp including $\mathbf{A}_{\mathsf{rec}} \in \mathbb{Z}_q^{n\times m}$ by challenger. Then, under its whole control, the adversary generates public keys $\mathsf{pk}_{\mathsf{OA}} = (\mathbf{U}_{\mathsf{oa},1}, \mathbf{U}_{\mathsf{oa},2}) \in (\mathbb{Z}_q^{n\times m})^2$ and $\mathsf{pk}_{\mathsf{GM}}$, and triggers the JOIN protocol with the challenger to register and certify the public key $\mathsf{pk}_j = (\mathbf{U}_{j,1}, \mathbf{U}_{j,2}) \in (\mathbb{Z}_q^{n\times m})^2$ for some honest receiver of the challenger's choice. After that, the adversary $\mathcal{A}$ makes a polynomial number of queries to DEC oracle which is faithfully handled by the challenger using $\mathbf{E}_{j,1}$. Then, $\mathcal{A}$ provides a valid witness $\mathbf{m} \in \{0,1\}^m$ satisfying the $\mathsf{Prohibitive}$ for challenge. Subsequently, the challenger take $b = 1$ and computes a ciphertext $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ which contains a group encryption of the real plaintext $\mathbf{m}$ under pk_j and returns it back as a challenger ciphertext. Then, a polynomial number of real proofs $\pi^*_{\Psi^*}$ which are associated with the challenge ciphertext Ψ^* are followed, and the decryption oracle with obvious restrictions is further granted. After doing this, $\mathcal{A}$ halts this game and outputs its guess bit $b' \in \{0,1\}$.

Game 2: This game is identical to Game 1 except one change in handling the ciphertext decryption oracle $\mathsf{DEC}(\mathsf{sk}_j, .)$. Concretely, $\mathcal{B}$ uses $\mathbf{E}_{j,2} \in \mathbb{Z}_q^{m\times m}$ instead of $\mathsf{sk}_j = \mathbf{E}_{j,1} \in \mathbb{Z}_q^{m\times m}$ to decrypt $\mathbf{c}_{\mathsf{rec}}$ among the ciphertext $\Psi = (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}})$. In the $\mathcal{A}$'s view, this game is the same as Game 1 until the event F_3 that $\mathcal{A}$ queries a ciphertext $\Psi = (\mathbf{c}_{\mathsf{rec},1}, \mathbf{c}_{\mathsf{rec},2}, \mathbf{c}_{\mathsf{oa}})$ where $\mathbf{c}_{\mathsf{rec},1}, \mathbf{c}_{\mathsf{rec},2}$ encrypts two distinct m-size messages. By the soundness of our argument presented in Sect. 4.3, $\Pr[W_2] - \Pr[W_1]$ is bounded by $\Pr[F_3] \leq \mathbf{Adv}^{\mathsf{sound}}_{\mathcal{B}}(\lambda)$.

Game 3: This game is like Game 2 except a modification in generating proofs $\pi^*_{\Psi^*}$. Instead of employing the real random coins $coins^*_{\Psi} = (\{\mathbf{s}^*_{\mathsf{rec},i}\}_i, \{\mathbf{x}^*_{\mathsf{rec},i}\}_i, \{\mathbf{y}^*_{\mathsf{rec},i}\}_i, \{\mathbf{s}^*_{\mathsf{oa},i}\}_i, \{\mathbf{x}^*_{\mathsf{oa},i}\}_i, \{\mathbf{y}^*_{\mathsf{oa},i}\}_i)$ used for Ψ^* to generate proofs, we rather to apply the zero-knowledge simulator presented in Sect. 4.3 once invoking $\mathsf{PROVE}^{b}_{\mathcal{P},\mathcal{P}'}$ after the challenge phase (i.e., given trusted public parameters, the computationally indistinguishable simulation is achieved with the techniques [17]). Here the computational ZK property ensures that, for any PPT adversary, the change is unnoticed: $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$.

Game 4: In this game, we modify the generation of $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ with $\mathbf{c}^*_{\mathsf{rec}} = (\mathbf{c}^*_{\mathsf{rec},1}, \mathbf{c}^*_{\mathsf{rec},2})$ by encrypting a random size-m message $\mathbf{m}' \in R_{\mathsf{pro}}$ as $\mathbf{c}^*_{\mathsf{rec},1}$, while still retaining $\mathbf{c}^*_{\mathsf{rec},2}$ for the encryption of $\mathbf{m} \in R_{\mathsf{pro}}$. By the semantic security of GPV dual encryption [18] (under the hardness assumption of the LWE problem) for public key $\mathsf{pk}_j = (\mathbf{U}_{j,1}, \mathbf{U}_{j,2})$, this game is identical to Game 3, i.e., $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

Game 5: This game makes one change by switching back to the application of $\mathbf{E}_{j,1} \in \mathbb{Z}_q^{m\times m}$ for the $\mathsf{DEC}(\mathsf{sk}_j, \cdot)$ queries with discarding $\mathbf{E}_{j,2}$, and the modification is invariant to the adversary except the event F_4, where the queries to the DEC for a valid ciphertext Ψ containing $\mathbf{c}^*_{\mathsf{rec},1}, \mathbf{c}^*_{\mathsf{rec},2}$ encrypting distinct messages satisfied the R_{Pro} relation, happens. But, the occurrence of F_4 implies that the simulation soundness of the underlying ZKAoK system used to generate Π_{GE} is broken. This results into $|\Pr[W_5 = 1] - \Pr[W_4 = 1]| \leq \mathbf{Adv}^{\mathsf{sound}}_{\Pi_{\mathsf{GE}}}(\lambda) = \mathsf{negl}(\lambda)$.

Game 6: Here, this experiment performs a modification to the Game 5 only by taking $\mathbf{c}_{\mathsf{rec},2}$ as the encryption of $\mathbf{m}' \in R_{\mathsf{Pro}}$ for the challenge ciphertext $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ with $\mathbf{c}^*_{\mathsf{rec}} = (\mathbf{c}^*_{\mathsf{rec},1}, \mathbf{c}^*_{\mathsf{rec},2})$. Note that this change is unnoticed to $\mathcal{A}$ due to the semantic security the encryption shares for public key $\mathbf{U}_{j,2}$, and also for the application of $\mathbf{E}_{j,1}$ to the DEC, we have $|\Pr[W_6 = 1] - \Pr[W_5 = 1]| = \mathsf{negl}(\lambda)$.

Game 7: Here, this experiment generates a real proof for ciphertext $\Psi^* = (\mathbf{c}^*_{\mathsf{rec}}, \mathbf{c}^*_{\mathsf{oa}})$ instead of using simulated proof, which is the only modification different to Game 6. The computational zero-knowledgeness of the underlying ZKAoK system makes the difference between Game 6 and Game 7 negligible, i.e., $\Pr[W_6 = 1] \approx \Pr[W_7 = 1]$. This is actually the experiment $\mathbf{Exp}^{\mathsf{sec}-0}_{\mathcal{A}}(\lambda)$, which directly leads that $\Pr[W_7 = 1] = \mathbf{Exp}^{\mathsf{sec}-0}_{\mathcal{A}}(\lambda)$. Thus, we have $|\mathbf{Exp}^{\mathsf{sec}-1}_{\mathcal{A}}(\lambda) - \mathbf{Exp}^{\mathsf{sec}-0}_{\mathcal{A}}(\lambda)| = \mathsf{negl}(\lambda)$, which proves the message security. □

Theorem 3. *The scheme is sound assuming that the* SIS *assumption holds.*

Proof. It suffices for us to prove these facts: for a given message filtering policy $\mathsf{Prohibitive}$, a ciphertext $\Psi^* = (\mathbf{c}_{\mathsf{rec}^*}, \mathbf{c}_{\mathsf{oa}^*})$, a Label L and an associated with proof Ψ^*, the public key associated with the identity revealed by the adversary is valid, certified, unique and the provided ciphertext Ψ^* is encrypted under this key. By the Lemma 2, the distribution of public keys is uniform, which ensures the public key is dense. In other words, the public is valid. In addition, the public key is unique since an occurring collision breaks the injective property of the mapping $\mathbf{F} \cdot [\mathbf{a}_1^\top | \mathbf{a}_2^\top]$. Thus, we only need to prove the other two cases.

a. The public key is certified (activated). If not, for some $j \in [0, N-1]$, there is an associated binary vector $\mathbf{p}_j \neq \mathbf{0}$ being accumulated into the published root value $\mathbf{u}$, but it is not equal to any value $\mathsf{bin}(\mathbf{F} \cdot [\mathbf{a}_1^\top | \mathbf{a}_2^\top])$, which contradicts the security of the accumulator.

b. The ciphertext is actually an encryption of witness $\mathbf{m}$ under this public key. If not, this event implies a breach in the computational soundness of our argument system and the binding property of the commitment scheme, which breaks the assumed hardness of the SIS problem. □

Efficiency Analysis. It can be seen that all algorithms used for the construction of the present group encryption are polynomially effective. The efficiency evaluation of the scheme is shown as follows.

– The public key of GM is a vector with bit-size $\widetilde{\mathcal{O}}(\lambda)$, and that of OA and users are respectively a matrix of bit-size $\widetilde{\mathcal{O}}(\lambda^2)$.
– The GM's secret key is given by a bit string of size $\widetilde{\mathcal{O}}(\lambda)$, and the secret keys of OA and users are respectively a small-norm matrix of bit size $\widetilde{\mathcal{O}}(\lambda^2)$.
– The ciphertext Ψ consists of $\mathbf{c}_{\mathsf{rec}} = (\mathbf{c}_{\mathsf{rec},1}, \mathbf{c}_{\mathsf{rec},2}) \in (\mathbb{Z}_q^m \times \mathbb{Z}_q^m)^2$ and $\mathbf{c}_{\mathsf{oa}} = (\mathbf{c}_{\mathsf{oa},1}, \mathbf{c}_{\mathsf{oa},2}) \in (\mathbb{Z}_q^m \times \mathbb{Z}_q^\ell)^2$, which leads the total bit size $\widetilde{\mathcal{O}}(\lambda + \ell)$.
– The communication cost of the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ largely relies on the bit-size of witness $\mathbf{x}$ with size $n_2 = m + 2n + 2m(k + nk + 2\delta_{2B}) + (m - t + 1)e\delta_{m-d}$ shown in Sect. 4.3, which leads $\widetilde{\mathcal{O}}(\lambda^2)$ bit-size.

In Table 1, given a security parameter λ, let $N = 2^\ell$, κ and Σ be the group size, the number of protocol repetitions and a one-time signature, respectively, we give a somewhat rough comparison between our scheme and the currently existing post-quantum secure group encryption schemes [25] (lattice-based variant) and [35] (code-based variant) in terms of functionality, efficiency and security. In the solid security, the full dynamicity is achieved with a highly reasonable cost: the GM only needs to update values of size $\widetilde{\mathcal{O}}(\ell\lambda)$ when group information changes.

Table 1. Comparison between schemes [25,35] and ours

Scheme	GM		OA		U		Ciph.	Commu.	Dynam.	Model
	pk	sk	pk	sk	pk	sk				
[25]	$\widetilde{\mathcal{O}}(\ell\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda) + \lvert\Sigma\rvert$	$\kappa\widetilde{\mathcal{O}}(\lambda^2)$	partial	Std.
[35]	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\kappa\widetilde{\mathcal{O}}(\lambda^2)$	full	RO.
Ours	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda^2)$	full	Std.

To better understand the advantage of our design, we also give a slightly concrete efficiency comparison between our scheme and the post-quantum safe schemes [25] and [35] for a same group size $N = 2^{10}$ toward the 80-bit security. By using the security analysis techniques shown in [35,43, and references therein], we choose the trade-off parameters as $(n, q) = (2795, 1125899906842679 \approx 2^{50})$, $(n, k_1, t_1, k_2, t_2, m, t_m, p, t, k) = (8192, 7997, 7, 7711, 18, 2^{38}, 279, 1024, 64, 10)$ and $(n, q, t, e, d) = (222, 524309 \approx 2^{19}, 64, 10, 10)$ for these schemes and ours, respectively. The results are shown in Table 2 where all the sizes of keys, ciphertexts and communication cost are almost highly superior than those of previous schemes. Particularly, our scheme obtains the drastic efficiency gains compared to [25] due to the free-of-trapdoor design. Besides, the group update cost of [35] and ours is 10.00 KB and 5.15 KB, respectively.

Table 2. Efficiency comparison between schemes [25,35] and ours

	GM		OA		U		Ciph.	Commu.
	pk	sk	pk	sk	pk	sk		
[25]	68.60 GB	482.55 GB	2.37 GB	38.86 GB	2.37 GB	38.86 GB	2.36 TB	3728 TB
[35]	1.00 KB	32.00 GB	15.62 MB	46.86 MB	15.06 MB	45.24 MB	4.00 KB	66107 TB
Ours	0.54 KB	1.08 KB	10.85 KB	129.50 KB	9.40 MB	112.30 MB	0.13 MB	10.32 GB

6 Conclusion

In this paper, we provide a re-formalized definition and security model of FDGE that is essentially equal to but more succinct than that of [35]. Then, we provide two generic and efficient zero-knowledge proof methods for demonstrating the inequalities of binary vectors, which can be readily extended to the case of general vectors. Finally, combining the appropriate cryptographic materials and the proof techniques just presented, we achieve the first lattice-based group encryption system which meanwhile offers the full dynamicity and the message filtering policy. Our scheme is constructed in a simpler manner and nearly outweighs the post-quantum secure ones [25,35] in terms of functions, efficiency and security.

Acknowledgement. This work has been supported by National Cryptography Development Fund (No. MMJJ20180110), National Natural Science Foundation of China (No. 61960206014), (No. 62121001) and (No. 61972429), and Guangdong Major Project of Basic and Applied Basic Research (No. 2019B030302008).

References

1. El Aimani, L., Joye, M.: Toward practical group encryption. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 237–252. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38980-1_15
2. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_1
3. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen **296**(1), 625–635 (1993)
4. Barić, N., Pfitzmann, B.: Collision-free accumulators and fail-stop signature schemes without trees. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 480–494. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_33
5. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 368–385. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_20
6. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38

7. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: the case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_11
8. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: CCS, pp. 168–177. ACM (2004)
9. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 117–136. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_7
10. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC, pp. 575–584. ACM (2013)
11. Bresson, E., Stern, J.: Efficient revocation in group signatures. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 190–206. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_15
12. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_5
13. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_20
14. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
15. Cathalo, J., Libert, B., Yung, M.: Group encryption: non-interactive realization in the standard model. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 179–196. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_11
16. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
17. Damgård, I.: Efficient concurrent zero-knowledge in the auxiliary string model. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 418–430. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_30
18. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206. ACM (2008)
19. Izabachène, M., Pointcheval, D., Vergnaud, D.: Mediated traceable anonymous encryption. In: Abdalla, M., Barreto, P.S.L.M. (eds.) LATINCRYPT 2010. LNCS, vol. 6212, pp. 40–60. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14712-8_3
20. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_34
21. Kiayias, A., Tsiounis, Y., Yung, M.: Group encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 181–199. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_11
22. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. Int. J. Secur. Netw. **1**(1/2), 24–45 (2006)
23. Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk, H. (ed.) PKC 2014. LNCS,

vol. 8383, pp. 345–361. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_20

24. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 373–403. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_13
25. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 101–131. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_4
26. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 1–31. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_1
27. Libert, B., Ling, S., Nguyen, K., Wang, H.: Lattice-based zero-knowledge arguments for integer relations. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10992, pp. 700–732. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_24
28. Libert, B., Peters, T., Yung, M.: Scalable group signatures with revocation. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 609–627. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_36
29. Libert, B., Yung, M., Joye, M., Peters, T.: Traceable group encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 592–610. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_34
30. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 107–124. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_8
31. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-based group signatures: achieving full dynamicity with ease. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 2017. LNCS, vol. 10355, pp. 293–312. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61204-1_15
32. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_2
33. Nakanishi, T., Fujii, H., Hira, Y., Funabiki, N.: Revocable group signature schemes with constant costs for signing and verifying. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 463–480. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00468-1_26
34. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: ACM, pp. 427–437. ACM (1990)
35. Nguyen, K., Safavi-Naini, R., Susilo, W., Wang, H., Xu, Y., Zeng, N.: Group encryption: full dynamicity, message filtering and code-based instantiation. In: Garay, J.A. (ed.) PKC 2021. LNCS, vol. 12711, pp. 678–708. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75248-4_24
36. Nguyen, L.: Accumulators from bilinear pairings and applications. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 275–292. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_19

37. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
38. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342. ACM (2009)
39. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93. ACM (2005)
40. Sakai, Y., Schuldt, J.C.N., Emura, K., Hanaoka, G., Ohta, K.: On the security of dynamic group signatures: preventing signature hijacking. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 715–732. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_42
41. Stern, J.: A new paradigm for public key identification. IEEE Trans. Inf. Theory **42**(6), 1757–1768 (1996)
42. Trolin, M., Wikström, D.: Hierarchical group signatures. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 446–458. Springer, Heidelberg (2005). https://doi.org/10.1007/11523468_37
43. Yang, R., Au, M.H., Zhang, Z., Xu, Q., Yu, Z., Whyte, W.: Efficient lattice-based zero-knowledge arguments with standard soundness: construction and applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 147–175. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_6