Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026820
Article

Tech Science Press

Policy-Based Group Signature Scheme from Lattice

Yongli Tang[1], Yuanhong Li[1], Qing Ye[1,*], Ying Li[1] and Xiaojun Wang[2]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China
[2]School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
*Corresponding Author: Qing Ye. Email: yeqing@hpu.edu.cn
Received: 05 January 2022; Accepted: 23 February 2022

Abstract: Although the existing group signature schemes from lattice have been optimized for efficiency, the signing abilities of each member in the group are relatively single. It may not be suitable for complex applications. Inspired by the pioneering work of Bellare and Fuchsbauer, we present a primitive called policy-based group signature. In policy-based group signatures, group members can on behalf of the group to sign documents that meet their own policies, and the generated signatures will not leak the identity and policies of the signer. Moreover, the group administrator is allowed to reveal the identity of signer when a controversy occurs. Through the analysis of application scenarios, we concluded that the policy-based group signature needs to meet two essential security properties: simulatability and traceability. And we construct a scheme of policy-based group signature from lattice through techniques such as commitment, zero-knowledge proof, rejection sampling. The security of our scheme is proved to be reduced to the module short integer solution (MSIS) and module learning with errors (MLWE) hard assumptions. Furthermore, we make a performance comparison between our scheme and three lattice-based group signature schemes. The result shows that our scheme has more advantages in storage overhead and the sizes of key and signature are decreased roughly by 83.13%, 46.01%, respectively, compared with other schemes.

Keywords: Group signature; policy-based signature; lattice-based cryptography; zero-knowledge proof

1 Introduction

1.1 Policy-Based Signature

Policy-based signature (PBS) is a novel concept of digital signature, which was proposed by Bellare et al. [1] at PKC 2014. PBS requires that signer can only sign documents that satisfy certain policy conditions. The users that do not satisfy the policy conditions cannot possess the ability of legitimate signers, and the signatures will not leak the identity and policy of signers. In [1] introduced two strong security notions: simulatability and extractability. The simulatability means that a legitimate signature is indistinguishable from a simulated signature, which is generated by a signature simulator

that does not need signing key or policy; the extractability means that there is an extractor, which is able to extract information of policy and identity from a legitimate signature, but cannot extract from forged signatures generated by an attacker. The simulatability and extractability are strong forms of indistinguishability, and unforgeability respectively according to [1]. With the two security notions, PBS will be effectively applied in hierarchical environment. For instance, in an enterprise, the authority expects that the employees in different departments or positions to have different signing abilities. Specifically, the employees in research department can only sign documents related to the research, and employees in finance department can only sign documents related to the finance. In 2016, Cheng et al. [2] constructed a scheme of PBS from lattice assumptions based on a zero-knowledge argument system and Bonsai tree.

1.2 Group Signature

Group signatures (GS) was proposed by Chaum et al. [3], which is an important cryptographic primitive. In GS, legitimate group members can represent the group to sign documents anonymously (anonymity); and the group administrator is allowed to open a signature by the tracking key to obtain the identity of signer (traceability). Due to the two properties of anonymity and traceability, GS can be applied in a variety of scenarios, such as e-commerce systems, trusted computing platforms, electronic voting, and much more.

In recent years, with the breakthroughs in quantum research, GS schemes based on hard assumptions of lattice have attracted the attention of scholars. In 2010, Gordon et al. [4] designed the first GS scheme from lattice in random oracle model (ROM) by the technology of GPV trapdoor, as well as the anonymity and traceability of the scheme can be reduced to the hard assumptions of learning with errors (LWE) and GapSVP respectively. But the storage overhead of keys and signature of their scheme is relatively large, which is linear with the number of group members. In 2013, Laguillaumie et al. [5] constructed a GS scheme with logarithmic size based on the non-interactive zero-knowledge proof of knowledge (NIZKPoK) under the hard assumptions of short integer solution (SIS) and LWE. Since then, a series of GS from lattice based on NIZKPoK have been proposed [6–10], and their storage cost has reached logarithmic size. Later, the constant-size GS are constructed by Ling et al. [11] and Zhang et al. [12]. The former is based on the "confined guessing" technique of Ducas et al. signature scheme [13]; and the latter uses a compact and scalable identity-encoding technique. Their schemes make the storage cost of keys and signatures independent of the number of group members. However, the NIZKPoK in the above GS schemes needs enough parallel repetition during execution due to its soundness error. This will cause a large cost of parameter and time so that the size of the keys and signature is still large, although it is independent of the number of group members. Therefore, Pino et al. [14] designed a new zero-knowledge proof protocol based on the signature scheme of [15] under the hard assumptions of MSIS and MLWE. Since this protocol limits the size of the message and challenge space, it has a smaller cost of parameter and time compared to other zero-knowledge proof protocols. The GS scheme based on this protocol also has more advantages in the storage overhead of the key and signature. Similarly, Boschini et al. [16] constructed a floppy-sized GS scheme by relaxed zero-knowledge proofs under the hard assumptions of ring short integer solution (RSIS) and ring learning with errors (RLWE). In 2019, a GS scheme without NIZK from lattice was designed by Katsumata et al. [17], but this construction requires a combination of attribute-based encryption and signatures. In 2020, Sun et al. [18] and Canard et al. [19] designed an improved scheme based on [17]. In conclusion, with the deepening of research in the field of GS from lattice, the size of GS has been effectively reduced. However, the above GS schemes from lattice are just be applied in the scenarios where the signing capabilities of the group members are relatively consistent. However, the different

signing capabilities of each group member are necessary for the GS scheme in actual scenarios, i.e., enterprises involving multiple departments, electronic voting for multiple regions, and much more. Therefore, GS requires a new primitive to be suitable for more extensive scenarios.

1.3 Our Contributions

In this work, we will define a concept of policy-based group signature (PBGS) based on previous work. Consider the following simple situation: Alice, Bob and Carol are employees of a company. The former two are from the research department and the latter is from the finance department. The authority of the company wants to develop a policy, which Alice, Bob and Carol are only allowed to sign documents that only related to their own department. At the same time, the signatures they generate can represent the company, and the identity of the signer will not be leaked. But if one day a document related to the research department causes a dispute (assuming Bob is the actual signer), the administrator of the company should be allowed to recover the identity of the signer (Bob) by the tracking key. In the above case, Alice, Bob and Carol are required to have different signing capabilities. Thus, the previous GS are not suitable. However, for PBGS, the authority wishes that Alice, Bob and Carol will be distributed signing keys and policies related to the department so that their signing capabilities will differ depending on the policy. The group member will not be able to sign when his policy does not satisfy some relationship with the document to be signed (unforgeability); Alice, Bob, Caro, or other outsiders of the company are unable to know the identity of the signer from a signature (simulatability). Even if given a signature related to the finance department, of which Carol is the only one employee, Carol is still anonymous due to the distribution of policies is a secret. The identity of the signer will be recovered through the PBGS administrator by the tracking key (traceability). In conclusion, the PBGS scheme in the application scenario needs to meet the following security requirements: simulatability, unforgeability and traceability. However, according to the definition of [20], unforgeability is unnecessary for GS because traceability has implied unforgeability. The same is true for the extractability defined in PBS. Therefore, we have extracted two security properties for PBGS: simulatability and traceability. With the above two security properties, PBGS will be applied in a wide range of fields. In addition to the enterprises involving multiple departments, the application of PBGS also includes hierarchical electronic voting for multiple regions, digital copyright management, and much more [21–23].

We show a construction of policy-based group signatures from lattice for the above primitive of PBGS, and it can resist the attacks of existing quantum algorithms. Our scheme satisfies the simulatability and full traceability in ROM under the security model of PBGS defined in Section 3.2. And the simulatability and full traceability are proved to be reduced to MLWE and MSIS assumptions, respectively. In terms of efficiency analysis, our scheme is compared with the three schemes of GS from lattice [11,16,17] in storage overhead. The analysis results show that the storage costs of our scheme are totally independent of the number of group members. The size of the key and signature are of order $\tilde{O}(\lambda)$. Specifically, the size of the signature under a set of practical parameters is decreased roughly by 46.01% on average compared to the schemes of [11,16,17]. And the size of keys also decreased roughly by 83.13%.

1.4 Our Techniques

At a high level, our PBGS scheme follows a template similar but not identical to the conventional GS defined by Bellare et al. [20]. In conventional GS, the public key, master key and traceability key are generated during the setup phase. But for PBGS, the policy relation also needs to be established to limit the signing ability of group members in the initial phase. After that, the key generation center (KGC)

will distribute the policy and the signing key to the group members. During the signature generation process, an efficient NIZKPoK about policy and signing keys is generated by group members. But if the policy of group members cannot satisfy the policy relation with the message to be signed, the signature algorithm will not be executed. Finally, in order to ensure full traceability, a verifiable encryption for identity will be generated by the group members. And then, the group administrator is allowed to decrypt the identity of the signer by the tracking key.

Specifically, we first review the requirements of policy language defined by [2]: (1) the space of message $\boldsymbol{M}$ should be large enough, and the space of policy $\boldsymbol{p}$ could be relatively small; (2) a policy $\boldsymbol{p}$ may simultaneously satisfy a lot of messages $\boldsymbol{M}$; (3) a message $\boldsymbol{M}$ could possibly satisfy a lot of policies $\boldsymbol{p}$. An instantiation for the above requirements of policy relation is constructed by Cheng et al. [2]. In particular, given a positive integer ℓ,n,d, if a signer with the policy $\boldsymbol{p} \in \{0,1\}^{\ell}$ is allowed to sign a message $\boldsymbol{M} \in \mathbb{Z}_2^n$, there is a witness $\boldsymbol{w} \in \{0,1\}^d$ satisfying $\boldsymbol{G}_1 \cdot \boldsymbol{p} + \boldsymbol{G}_2 \cdot \boldsymbol{w} = \boldsymbol{M} \pmod 2$, where$n - \ell < d$, $\boldsymbol{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ is a uniform random matrix, and $\boldsymbol{G}_2 \in \mathbb{Z}_2^{n \times d}$ is an approximate identity matrix. We define the relation as: $\mathrm{PR}(\{0,1\}^{\ell} \times \mathbb{Z}_2^n) \times \{0,1\}^d \to \{0,1\}$. That is,

$$\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 1 \Leftrightarrow \boldsymbol{G}_1 \cdot \boldsymbol{p} + \boldsymbol{G}_2 \cdot \boldsymbol{w} = \boldsymbol{M} \bmod 2.$$

It satisfies the above requirements of the policy language, and its hardness is based on the LWE hard assumption.

In the signature generation phase, the signer needs to possess policy $\boldsymbol{p}$, witness $\boldsymbol{w}$, and signing key $\boldsymbol{s}$ in order to sign a message $\boldsymbol{M}$ that satisfies the policy $\boldsymbol{p}$, among which the signing key $\boldsymbol{s}$ is obtained by preimage sampling introduced in [24]. Specifically, we first generate a trapdoor $\boldsymbol{R}$ in the setup phase. After that, $\boldsymbol{s}$ is obtained through preimage sampling algorithm, which is executed by KGC through inputting parameters such as the policy $\boldsymbol{p}$, the identity of signer and the system public key. Then, $\boldsymbol{s}$ and $\boldsymbol{p}$ constitute a secret pair $(\boldsymbol{p}, \boldsymbol{s})$. At the moment, the two facts about the secret pair $(\boldsymbol{p}, \boldsymbol{s})$ and the policy relation have been possessed for the signer. In order to convince the verifier, the signer needs to generate a NIZKPoK about the linear relation for the two facts. The technically challenging question is that policy $\boldsymbol{p}$ satisfies two relations at the same time. Hence it is the key to construct a suitable proof protocol. We will show a new proof protocol based on the linear relation proof from [14] to prove the above facts, and it will be applied to our PBGS scheme after Fiat-Shamir transformation. Furthermore, in order to ensure the full traceability, we will integrate an efficient commitment technology from [25] to generate commitment $Com(i, \boldsymbol{r})$ about the signer's identity i and a random $\boldsymbol{r}$ during the signing process. Then the random $\boldsymbol{r}$ will be encrypted by the technology of verifiable encryption from [26]. And the ciphertext and the transcript of the above NIZKPoK will be formed a signature, which will be verified in the verification algorithm. After that, the group administrator can obtain $\boldsymbol{r}$ through using the tracking key to decrypt the ciphertext, and then open the commitment $Com(i, \boldsymbol{r})$ to obtain the signer's identity i.

2 Preliminaries

2.1 Symbol Definition

The symbols that appear in this article are described in Tab. 1.

Table 1: Symbol definition

Notations	Description
$\mathbb{Z}_q^{n\times m}$	$n \times m$ dimensional matrix of modulo q residue class ring
R_q	Modulo q polynomial ring of$R_q = \mathbb{Z}_q[x]/(x^d + 1)$
$\boldsymbol{A}$	Matrix
$\boldsymbol{x}$	Column vector
$\|\|\boldsymbol{x}\|\|$	Euclidean norm of vector$\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|_\infty$	Infinite norm of vector$\boldsymbol{x}$
$\boldsymbol{x} \leftarrow D$	Chooses vector from $\boldsymbol{x}$ probability distribution$D$

2.2 MSIS and MLWE

Definition 1 ($\text{MSIS}_{l,m,\beta}$ [27]) Given parameters l, m, β and $\boldsymbol{A} \in R_q^{l\times m}$, the $\text{MSIS}_{l,m,\beta}$ is defined as: Finding $\boldsymbol{z} \in R^m$ such that $\boldsymbol{Az} = \boldsymbol{0}$ and $0 < ||\boldsymbol{z}||_\infty \leq \beta$.

Lemma 1 [27] For any$\beta = \text{poly}(d), m \geq 1, \varepsilon > 0, \gamma \geq \beta\sqrt{d \cdot l}\cdot\omega(\sqrt{\log d \cdot l})\cdot\sqrt{\ln(2d(1 + 1/\varepsilon)/\pi)}$ and $q \geq \beta\sqrt{d \cdot m}$, $\text{MSIS}_{l,m,\beta}$ is as difficult as the SIVP_γ problem at least.

Definition 2 ($\text{MLWE}_{m,n,\chi}$ [27]) Given parameters m, n and error distribution$\chi = \{a \in R, ||a||_\infty \leq 1\}$. For $(\boldsymbol{s}, \boldsymbol{e}) \leftarrow \chi^n \times \chi^m$ and $\boldsymbol{A} \leftarrow R_q^{m\times n}$, the $\text{MLWE}_{m,n,\chi}$ is defined as: Distinguishing samples chosen from $(\boldsymbol{A}, \boldsymbol{As} + \boldsymbol{e})$ and samples chosen from uniform distribution $(\boldsymbol{A}, \boldsymbol{b}) \leftarrow R_q^{n\times m} \times R_q^n$.

Lemma 2 [27] For $m, n > 0, \alpha \in (0, 1), \varepsilon > 0, \gamma = \sqrt{8d \cdot n} \cdot \omega(\sqrt{\log d}/\alpha) \cdot \sqrt{\ln(2d(1 + 1/\varepsilon)/\pi)}$ and $q \geq 2$, the $\text{MLWE}_{m,n,\chi}$ is as difficult as the SIVP_γ problem at least.

As discussed in [28], the practical hardness of the above assumptions is not affected by the parameter m to resist known attacks. Therefore, the assumptions will be simply written $\text{MSIS}_{l,\beta}$ and $\text{MLWE}_{n,\chi}$ by omitting the m, where the l and n represent the module ranks for MSIS and MLWE, respectively.

2.3 Discrete Gaussian Distribution and Rejection Sampling

Given any $\sigma > 0$, vector $c \in \mathbb{R}$ and function $\rho_{\sigma,c}(x) = \exp(-\pi\frac{||x-c||^2}{\sigma^2})$. Then the Gaussian distribution $D_{\sigma,c}$ centered in c is described as:

$D_{\sigma,c}(x) = \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(\mathbb{Z})}$ where $\rho_{\sigma,c}(\mathbb{Z}) = \sum_{x\in\mathbb{Z}} \rho_{\sigma,c}(x)$.

We will simply write D_σ when $c = 0$. And if the polynomial $x \in R$, $x \leftarrow D_\sigma$ is defined as every coefficient of x obeying distribution D_σ.

Lemma 3 [14] For any $\sigma > 0$, positive integer n and $k > 0$, the following formulas holds:

(1) $\Pr[x \leftarrow D_\sigma : |x| > k\sigma] \leq 2e^{-k^2/2}$.

(2) $\Pr\left[\boldsymbol{x} \leftarrow D_\sigma^n : ||\boldsymbol{x}|| > \sigma\sqrt{2n}\right] < 2^{-n/4}$.

At EUROCRYPT 2012, Lyubasevsky introduced an algorithm of rejection sampling, which can be executed with a certain probability. The description is as follows:

Algorithm 1: Rej($\boldsymbol{z}, \boldsymbol{b}, \sigma$)

$u \leftarrow [0, 1)$
If $u > \frac{1}{3} \cdot \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{b}\rangle + ||\boldsymbol{b}||^2}{2\sigma^2})$ then
  return 0
else
  return 1
end if

Lemma 4 [14,29,30] For $V = \{\boldsymbol{v} \in R^n : ||\boldsymbol{v}|| < t\}$, $\boldsymbol{b} \in R^n$ and $\sigma \geq 11||\boldsymbol{b}||$, a procedure will be run by sampling $\boldsymbol{y} \leftarrow D_\sigma^n$ and outputs Rej($\boldsymbol{z} := \boldsymbol{y} + \boldsymbol{b}, \boldsymbol{b}, \sigma$). Then the probability of returning 1 in Algorithm 1 is within $1/3 + 2^{-100}$. And the statistical distance between the distribution of $\boldsymbol{z}$ and D_σ^n is within 2^{-100} when the Algorithm 1 outputs 1.

2.4 Trapdoor from Lattice

Lemma 5 [16,24,31] Given positive integer n, m, q, i, parameter $\sigma = q^{1/m} \cdot O(\sqrt{nd} + \sqrt{md})$, polynomial $\boldsymbol{A} \in 1R^{1\times n}$ and $\boldsymbol{R} \leftarrow \chi^{n\times m}$. Set the gadget matrix $\boldsymbol{g}^T = [1 \quad q^{1/m} \quad \ldots \quad q^{(m-1)/m}]$. Let $\boldsymbol{B} = \boldsymbol{AR} \in R^{1\times m}$, we will get a basis $\boldsymbol{S} \in \mathbb{Z}^{(n+m)d\times(n+m)}$ for $\Lambda^\perp = \{\boldsymbol{x} \in R^{n+m} | [\boldsymbol{A}|\boldsymbol{AR} + i\boldsymbol{g}^T] \cdot \boldsymbol{x} = \boldsymbol{0} \,(\mathrm{mod} q)\}$, which fulfills $||\tilde{\boldsymbol{S}}|| \leq (s_1(\boldsymbol{R}) + 1)\sqrt{\delta^2 + 1}$ after Gram-Schmidt orthogonalization, where $\delta = \lceil\sqrt{q}\rceil$ and $s_1(\boldsymbol{R})$ means maximal singular value of $\boldsymbol{R}$. And then for any polynomial vector $u \in R$, there is an algorithm SampleD($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, u, \sigma$), which is able to sample from distribution $D^\perp_{[\boldsymbol{A}|\boldsymbol{AR}+i\boldsymbol{g}^T],\sigma,u}$ with a certain probability.

2.5 Commitments

Definition 3 (Commitment [25]) Given challenge space $C = \{c : c \in R, ||c||_1 = \kappa, ||c||_\infty = 1\}$, public matrices $\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{A'}_1 \end{bmatrix} \in R_q^{n\times k}$, $\boldsymbol{A}_2 = \begin{bmatrix} \boldsymbol{0}^{l\times n} & \boldsymbol{I}_l & \boldsymbol{A'}_2 \end{bmatrix} \in R_q^{l\times k}$. For the message $\boldsymbol{m} \in R_q^l$ to be committed and the random $\boldsymbol{r} \leftarrow \chi^k$, an effective commitment will be generated as follows:

$$Com(\boldsymbol{m}, \boldsymbol{r}) = \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \boldsymbol{r} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}.$$

If the following equation holds:

$$c \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \boldsymbol{r} + c \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}, \text{ where } ||\boldsymbol{r}|| \leq B_{com} \text{ and } c \in \bar{C},$$

We call $(\boldsymbol{m}, \boldsymbol{r}, c)$ is a valid opening of commitment.

Lemma 6 [25] The above commitments have the following properties:

(1) (Binding) Let $\kappa \geq \max_{c\in C}(||c||_1)$, if an attacker $\mathcal{A}$ who has advantage ε in outputting a commitment through two valid $(\boldsymbol{m}, \boldsymbol{r}, c)$ and $(\boldsymbol{m'}, \boldsymbol{r'}, c')$ such that $\boldsymbol{m} \neq \boldsymbol{m'}$, there is an algorithm $\mathcal{A}'$ who has advantage ε in solving the $\mathrm{MSIS}_{n,4\kappa B_{Com}}$ within the same time.

(2) (Hiding) For $\boldsymbol{m}, \boldsymbol{m'} \in \boldsymbol{R}_q^l$, if an attacker $\mathcal{A}$ has advantage ε in distinguishing between $Com(\boldsymbol{m}, \boldsymbol{r})$ and $Com(\boldsymbol{m'}, \boldsymbol{r'})$, there is an algorithm $\mathcal{A}'$ that has advantage $\varepsilon/2$ in solving the $\mathrm{MLWE}_{k-n-l,\chi}$ in the same time.

The detailed proof of the above lemma could be found in the work [14,25].

3 Definition of Policy-Based Group Signature and Security Model

3.1 Definition

Definition 4 (PBGS) A policy-based group signature composed of five polynomial-time algorithms:

(1) GSetup (1^λ): It takes the security parameter λ as input, builds the policy relation $\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w})$ and outputs group public key gpk, group master private key gmk and administrator tracking key gtk.
(2) KeyGen (gmk,$\boldsymbol{p}$, i): It takes the group master private key gmk, policy $\boldsymbol{p}$ and member identity $i \in [N]$ as inputs, outputs a signing key $\mathrm{sk}_{\boldsymbol{p},i}$ of member i about the policy $\boldsymbol{p}$.
(3) Sign ($\mathrm{sk}_{\boldsymbol{p},i}$, $\boldsymbol{M}$, $\boldsymbol{w}$): It takes the signing key $\mathrm{sk}_{\boldsymbol{p},i}$, a message $\boldsymbol{M}$ and a witness $\boldsymbol{w}$ as inputs, outputs a signature $\sum$ if the policy relation satisfies $\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 1$, or $\perp$ otherwise.
(4) Verify (gpk, $\sum$, $\boldsymbol{M}$): It takes the group public key gpk, a signature $\sum$ and a message $\boldsymbol{M}$ as inputs, outputs "Valid" if the signature $\sum$ is a valid signature on message $\boldsymbol{M}$, or "Invalid" otherwise.
(5) Open (gtk, $\sum$): It takes the tracking key gtk and a signature $\sum$ as inputs, outputs the identity i of signer if the signature $\sum$ is "Valid" checked by algorithm Verify, or $\perp$ otherwise.

3.2 Security Model

A PBGS scheme should meet three security properties: correctness, simulatability and traceability. Correctness, is defined in Definition 5 detailedly, includes verification correctness and opening correctness. Simulatability implies that the attacker cannot confirm the identity of the signer through a signature because a valid signature is indistinguishable from a simulated signature. Please refer to Definition 6 for details. Traceability means that a valid signature should be opened through group administrator by the tracking key so that the identity of the signer is restored. Our scheme meets full traceability, which is defined in Definition 7 detailedly. Furthermore, anonymity and unforgeability could be unnecessary for PBGS. We will discuss this issue later in Section 3.3.

Definition 5 (Correctness) The correctness of the PBGS contains verification correctness and opening correctness. The verification correctness means that the probability of returning "Invalid" from the algorithm Verify is negligible for a signature generated honestly. That is:

$$\Pr\left[\text{"Invalid"} \leftarrow \mathrm{Verify}(\mathrm{gpk}, \sum, \boldsymbol{M}) \;\middle|\; \begin{array}{l} \mathrm{gpk}, \mathrm{gmk}, \mathrm{gtk} \leftarrow \mathrm{Gsetup}(1^\lambda) \\ \mathrm{sk}_{\boldsymbol{p},i} \leftarrow \mathrm{KeyGen}(\mathrm{gmk}, \boldsymbol{p}, i) \\ \sum \leftarrow \mathrm{Sign}(\mathrm{sk}_{\boldsymbol{p},i}, \boldsymbol{M}, \boldsymbol{w}) \end{array}\right] \leq negl(n).$$

The opening correctness means that the probability of returning $\perp$ from the algorithm Open is negligible for a signature generated honestly. That is:

$$\Pr\left[\perp \leftarrow \mathrm{Open}(\mathrm{gtk}, \sum) \;\middle|\; \begin{array}{l} \mathrm{gpk}, \mathrm{gmk}, \mathrm{gtk} \leftarrow \mathrm{Gsetup}(1^\lambda) \\ \mathrm{sk}_{\boldsymbol{p},i} \leftarrow \mathrm{KeyGen}(\mathrm{gmk}, \boldsymbol{p}, i) \\ \sum \leftarrow \mathrm{Sign}(\mathrm{sk}_{\boldsymbol{p},i}, \boldsymbol{M}, \boldsymbol{w}) \\ \text{"Valid"} \leftarrow \mathrm{Verify}(\mathrm{gpk}, \sum, \boldsymbol{M}) \end{array}\right] \leq negl(n).$$

Definition 6 (Simulatability) The simulatability requires that there is a simulator SimSign($\boldsymbol{M}$), which generates signatures without the need for any signing key or policy. Then the simulated signatures generated by SimSign($\boldsymbol{M}$) are indistinguishable from the signatures generated honestly. The simulatability game $\mathrm{Game}^{\mathrm{SIM}}_{\mathcal{PBGS},\mathcal{A}}(n)$ is defined by the following processes between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Setup: $\mathcal{C}$ runs the algorithm GSetup (1^{λ}) honestly by inputting the security parameter λ, and returns gpk and gmk to $\mathcal{A}$.

Queries: $\mathcal{A}$ is allowed to query adaptively the signing key for policy $\boldsymbol{p}$ and member $i \in [N]$, and $\mathcal{C}$ sends $\mathrm{sk}_{\boldsymbol{p},i}$ generated by running algorithm KeyGen(gmk, $\boldsymbol{p}$, i) to $\mathcal{A}$.

Challenge: $\mathcal{A}$ returns $i \in [N]$, $\boldsymbol{M}^*$ and $\boldsymbol{w}^*$. If $\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 0$, the game will be aborted. Otherwise, $\mathcal{C}$ computes $\sum_0 \leftarrow \mathrm{SimSign}(\boldsymbol{M}^*)$ and $\sum_1 \leftarrow \mathrm{Sign}(\mathrm{sk}_{\boldsymbol{p},i}, \boldsymbol{M}^*, \boldsymbol{w}^*)$. Then $\mathcal{C}$ selects random bit $b \in \{0, 1\}$ and returns $\sum_b$ to $\mathcal{A}$.

Finalization: $\mathcal{A}$ returns a guess $b' \in \{0, 1\}$. If $b' = b$, the game outputs 1.

The advantage of $\mathcal{A}$ in simulatability game is defined as:

$$\mathrm{Adv}^{\mathrm{SIM}}_{\mathcal{PBGS},\mathcal{A}}(n) = \left|\Pr\left[\mathrm{Game}^{\mathrm{SIM}}_{\mathcal{PBGS},\mathcal{A}}(n) \Rightarrow 1\right] - 1/2\right|.$$

If $\mathrm{Adv}^{\mathrm{SIM}}_{\mathcal{PBGS},\mathcal{A}}(n)$ is negligible, we call that the PBGS scheme meets simulatability.

Definition 7 (Full Traceability [20]) Full traceability is a strong form of traceability. It asks that a team of group members who concentrate their signing keys is unable to generate a valid signature, which could not be caught by the open algorithm. Even though the colluding group knows the tracking key of group manager, that is true. The full traceability game $\mathrm{Game}^{\mathrm{TRACE}}_{\mathcal{PBGS},\mathcal{A}}(n)$ is defined by the following processes between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Setup: $\mathcal{C}$ runs honestly the algorithm GSetup (1^{λ}) and initializes two lists Γ and I. Then $\mathcal{C}$ sends gpk and gtk to $\mathcal{A}$.

Queries: $\mathcal{A}$ have access to the following queries:

- Request for the signing key of member $i \in [N]$ and policy $\boldsymbol{p}$. $\mathcal{C}$ returns $\mathrm{sk}_{\boldsymbol{p},i} \leftarrow \mathrm{KeyGen}(\mathrm{gmk}, \boldsymbol{p}, i)$ to $\mathcal{A}$ and sets $\Gamma \leftarrow \Gamma \cup \{(\boldsymbol{p}, i)\}$.
- Request for the signature about any message $\boldsymbol{M}$ on identity i and policy $\boldsymbol{p}$. $\mathcal{C}$ returns $\sum_{\boldsymbol{M}} \leftarrow \mathrm{SimSign}(\boldsymbol{M})$ to $\mathcal{A}$ and sets $\mathrm{I} \leftarrow \mathrm{I} \cup \{(\boldsymbol{M}, \sum_{\boldsymbol{M}})\}$.

Finalization: $\mathcal{A}$ returns $(\boldsymbol{M}^*, \sum^*)$. If "Invalid" $\leftarrow \mathrm{Verify}(\mathrm{gpk}, \sum^*, \boldsymbol{M}^*)$ or $(\boldsymbol{M}^*, \sum^*) \in I$, the game outputs 0. Otherwise, $\mathcal{C}$ runs algorithm Open. The game outputs 1 if the algorithm Open returns $\perp$ or returns i, where $\{(\boldsymbol{p}, i)\} \notin \Gamma$. While in other cases, the game returns 0.

The advantage of $\mathcal{A}$ in full traceability game is written by:

$$\mathrm{Adv}^{\mathrm{TRACE}}_{\mathcal{PBGS},\mathcal{A}}(n) = \Pr\left[\mathrm{Game}^{\mathrm{TRACE}}_{\mathcal{PBGS},\mathcal{A}}(n) \Rightarrow 1\right].$$

If $\mathrm{Adv}^{\mathrm{TRACE}}_{\mathcal{PBGS},\mathcal{A}}(n)$ is negligible, we call that the PBGS scheme meets full traceability.

3.3 Discussion

As described in Section 1.3, the anonymity and unforgeability are unnecessary. First, the normal anonymity does not always provide the privacy for the policy relevant to the key and witness [1]. To see this, there is a policy relation such that for every message $\boldsymbol{M}$, only one policy $\boldsymbol{p}$ satisfies $\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 1$. In this situation, a scheme which is composed of the above policy relation still meets anonymity. But the policy is not hiding in this scheme. Indeed, the simulatability introduced by [1] requires that there is a simulator which is able to produce the simulated signatures does not need any signing key or policy, and the simulated signatures are indistinguishable from the signature generated honestly. Next, Traceability is a basic property for GS. It has implied the unforgeability of ordinary digital signatures according to the definition of [20] because the forgery game is a special case for the full-traceability

game. The same is true for the extractability game that PBS needs to have. Therefore, we say that the security attributes that PBGS needs to meet are simulation and traceability.

4 The Scheme

4.1 A ZKPoK Protocol

In this section, we present a ZKPoK protocol $\prod_{\mathcal{PBGS}}$ based on the linear relation proof from [14]. It will be used in the PBGS scheme and allows a prover to convince a verifier that he is a legitimate group member for a certain policy.

First, fix parameters $\lambda, \kappa, q, Q, \sigma$ and polynomial ring R (See our construction of PBGS in Section 4.2). For public information $\boldsymbol{A}, \boldsymbol{v}, \boldsymbol{G}_1, \boldsymbol{G}_2, u, \boldsymbol{t}, \boldsymbol{t}', \delta, \boldsymbol{B}, \boldsymbol{y}, \boldsymbol{M}, \boldsymbol{d}, h$ and secret information $(\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, \boldsymbol{w})$, the prover $\mathcal{P}$ will convince the verifier $\mathcal{V}$ that $\mathcal{P}$ possesses the secret $(\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, \boldsymbol{w})$ satisfying policy relation $\mathrm{PC}_{G_1,G_2}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 1$. Therefore, the protocol $\prod_{\mathcal{PBGS}}$ we will present should be able to prove the following facts:

- $(\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2})$ is a valid secret pair.
- $\boldsymbol{G}_1 \cdot \boldsymbol{p} + \boldsymbol{G}_2 \cdot \boldsymbol{w} = \boldsymbol{M}$.
- $(\boldsymbol{d}, h)$ is a verifiable ciphertext.

The interaction between the two parties is as follows:

Protocol 1: Zero-knowledge Protocol of Knowledge for PBGS

Protocol 1:

1. **Commitment.** $\mathcal{P}$ performs the following steps:

Selects $(\boldsymbol{y}_r, \boldsymbol{y}'_r) \leftarrow D^3_{\xi_1} \times D^3_{\xi_1}, \boldsymbol{y}_B \leftarrow D^8_{\xi_1}, (\boldsymbol{y}_{s_1}, \boldsymbol{y}_{s_2}) \leftarrow D^2_{\xi_2} \times D^2_{\xi_2}, \boldsymbol{y}_{s_3} \leftarrow D^3_{\xi_3}, (\boldsymbol{y}_p, \boldsymbol{y}_w) \leftarrow D^3_{\xi_1} \times D^{\ell-3}_{\xi_1}$. $\boldsymbol{y}_s = (\boldsymbol{y}_{s_1}, \boldsymbol{y}_{s_2}, \boldsymbol{y}_{s_3})^T$.

Computes $\boldsymbol{w}_1 = \boldsymbol{a}_1^T \boldsymbol{y}_r$, $\boldsymbol{w}'_1 = \boldsymbol{a}_1^T \boldsymbol{y}'_r$, $\boldsymbol{w}_2 = \delta \boldsymbol{a}_2^T \boldsymbol{y}_r - \boldsymbol{a}_2^T \boldsymbol{y}'_r$, $\boldsymbol{w}_s = \boldsymbol{v}^T \boldsymbol{y}_s$, $\boldsymbol{w}_B = \boldsymbol{B}\boldsymbol{y}_B$, $\boldsymbol{w}_p = \boldsymbol{G}_1 \boldsymbol{y}_p + \boldsymbol{G}_2 \boldsymbol{y}_w$.

Sends $\boldsymbol{w}_1, \boldsymbol{w}'_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p$ to $\mathcal{V}$.

2. **Challenge.**

$\mathcal{V}$ generates a challenge $c \leftarrow C$ and sends c to $\mathcal{P}$.

3. **Response.**

$\mathcal{P}$ computes $\boldsymbol{z} = \boldsymbol{r}c + \boldsymbol{y}_r$, $\boldsymbol{z}' = \boldsymbol{r}'c + \boldsymbol{y}'_r$, $\boldsymbol{z}_{s_1} = \boldsymbol{s}_1 c + \boldsymbol{y}_{s_1}$, $\boldsymbol{z}_{s_2} = \boldsymbol{s}_2 c + \boldsymbol{y}_{s_2}$, $\boldsymbol{z}_{s_3} = (\boldsymbol{p} - \begin{bmatrix} \boldsymbol{r} & \boldsymbol{r}' \end{bmatrix} \boldsymbol{s}_2)c + \boldsymbol{y}_{s_3}$, $\boldsymbol{z}_B = \boldsymbol{r}_B c + \boldsymbol{y}_B$, $\boldsymbol{z}_p = \boldsymbol{p}c + \boldsymbol{y}_p$, $\boldsymbol{z}_w = \boldsymbol{w}c + \boldsymbol{y}_w$.

Run rejection sampling $\mathrm{Rej}((\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_B, \boldsymbol{z}_p, \boldsymbol{z}_w)(\boldsymbol{r}c, \boldsymbol{r}'c, \boldsymbol{r}_B c, \boldsymbol{p}c, \boldsymbol{w}c), \xi_1)$, $\mathrm{Rej}((\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}), (\boldsymbol{s}_1 c, \boldsymbol{s}_2 c), \xi_2)$ and $\mathrm{Rej}(\boldsymbol{z}_{s_3}, \boldsymbol{s}_3 c, \xi_3)$, returns $\Pi(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_{s_3}, \boldsymbol{z}_p, \boldsymbol{z}_w, \boldsymbol{z}_B, c)$ to $\mathcal{V}$.

4. **Verification.**

$\mathcal{V}$ checks:

$$\begin{cases} \boldsymbol{a}_1^T \boldsymbol{z} = t_1 c + \boldsymbol{w}_1 \\ \boldsymbol{a}_1^T \boldsymbol{z}' = t'_1 c + \boldsymbol{w}'_1 \\ \delta \boldsymbol{a}_1^T \boldsymbol{z} - \boldsymbol{a}_2^T \boldsymbol{z}' = (\delta t_2 - t'_2)c + \boldsymbol{w}_2 \\ \boldsymbol{v}^T \boldsymbol{z}_s = uc + \boldsymbol{w}_s \\ \boldsymbol{B}\boldsymbol{z}_B = \boldsymbol{y}c + \boldsymbol{w}_B \\ \boldsymbol{G}_1 \boldsymbol{z}_p + \boldsymbol{G}_2 \boldsymbol{z}_w = \boldsymbol{M}c + \boldsymbol{w}_p \\ ||(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_B, \boldsymbol{z}_p, \boldsymbol{z}_w)|| \le B_1 \wedge ||(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2})|| \le B_2 \wedge ||\boldsymbol{z}_{s_3}|| \le B_3 \end{cases}$$

The verifier $\mathcal{V}$ returns 1 if all of the above equations hold, otherwise it returns 0.

Theorem 1 Given $\boldsymbol{r}, \boldsymbol{r}' \leftarrow D^3_\sigma$, $\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2} \leftarrow D^2_\sigma$, $\boldsymbol{p} \leftarrow \chi^3$, $\boldsymbol{w} \leftarrow \chi^{\ell-3}$ and $\boldsymbol{G}_1, \boldsymbol{G}_2, u, \boldsymbol{t}, \boldsymbol{t}', h, \boldsymbol{d}, \boldsymbol{B}, \boldsymbol{y}$ fixed in Section 4.2, for $\xi_1 \ge 11\kappa\sqrt{(14+\ell)d}$, $\xi_2 \ge 11\kappa\sqrt{4d}\sigma$, $\xi_3 \ge 11\kappa(d\sqrt{24}\sigma + \sqrt{2d}\sigma)$ and

$B_1 \geq \sqrt{2(14+\ell)d}\xi_1$, $B_2 \geq 2\sqrt{2d}\xi_2$, $B_3 \geq \sqrt{6d}\xi_3$, the protocol $\prod_{\mathcal{PBGS}}$ in Protocol 1 meets the following properties:

- Correctness: The prover $\mathcal{P}$ outputs successfully a transcript with a probability of $1/27 + 2^{-100}$ at least. And the verifier $\mathcal{V}$ will accept the transcript with overwhelming probability when the protocol is not aborted.
- Honest-Verifier Zero-Knowledge: An honest verifier can simulate the transcripts with statistically indistinguishable distribution when the protocol is not aborted.
- Special Soundness: A valid opening of commitment $\boldsymbol{t}, \boldsymbol{t}'$ can be extracted by two accepting transcripts.

Proof.

Correctness: If $\mathcal{P}$ is an honest prover, it can be got from Lemma 4 that the probability of rejection sampling is at least $1/27+2^{-100}$. The distribution $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_p, \boldsymbol{z}_w)$, $\boldsymbol{z}_B$ and $\boldsymbol{z}_{s_3}$ is close to $D^{14+\ell}_{\xi_1}$, $D^4_{\xi_2}$ and $D^3_{\xi_3}$ after the rejection sampling. And we can get $||(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_B, \boldsymbol{z}_p, \boldsymbol{z}_w)|| \leq B_1 \wedge ||(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2})|| \leq B_2 \wedge ||\boldsymbol{z}_{s_3}|| \leq B_3$ will be held with an overwhelming probability according to Lemma 3. Therefore, $\mathcal{V}$ will accept the transcript with overwhelming probability.

Honest-Verifier Zero-Knowledge: We only show that the protocol $\prod_{\mathcal{PBGS}}$ meets honest-verifier zero-knowledge when the prover $\mathcal{P}$ is not aborted. Since the protocol will be converted to NIZKPoK by Fiat-Shamir transformation and be applied to PBGS. $\mathcal{V}$ cannot get the transcript when the protocol is aborted. Then for a non-abort protocol, there is a probabilistic polynomial time (PPT) simulation algorithm $S(\boldsymbol{A}, \boldsymbol{v}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{B})$:

$c \leftarrow C$.

$\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_p, \boldsymbol{z}_w \leftarrow D_{\xi_1},\ \boldsymbol{z}_B \leftarrow D_{\xi_2},\ \boldsymbol{z}_{s_3} \leftarrow D_{\xi_3}$.

$\boldsymbol{w}_1 = \boldsymbol{a}_1^T\boldsymbol{z} - t_1 c,\ \boldsymbol{w'}_1 = \boldsymbol{a}_1^T\boldsymbol{z}' - t_1' c,\ \boldsymbol{w}_2 = \delta\boldsymbol{a}_2^T\boldsymbol{z} - \boldsymbol{a}_2^T\boldsymbol{z}' - (\delta t_2 - t_2')c$.

$\boldsymbol{w}_s = \boldsymbol{v}^T\boldsymbol{z}_s - uc,\ \boldsymbol{w}_B = \boldsymbol{B}\boldsymbol{z}_B - \boldsymbol{y}c,\ \boldsymbol{w}_p = \boldsymbol{G}_1\boldsymbol{z}_p + \boldsymbol{G}_2\boldsymbol{z}_w - \boldsymbol{M}c$.

Output $(\boldsymbol{w}_1, \boldsymbol{w'}_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p, c, \boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_{s_3}, \boldsymbol{z}_p, \boldsymbol{z}_w, \boldsymbol{z}_B)$.

We will get that the transcripts generated by the simulation algorithm $S(\boldsymbol{A}, \boldsymbol{v}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{B})$ will be accepted by the verifier with overwhelming probability. In the real protocol, the statistical distance between distribution of $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_p, \boldsymbol{z}_w)$, $\boldsymbol{z}_B$, $\boldsymbol{z}_{s_3}$ and distribution $D^{14+\ell}_{\xi_1}$, $D^4_{\xi_2}$, $D^3_{\xi_3}$ is no more than 2^{-100}. Since $\boldsymbol{w}_1, \boldsymbol{w'}_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p$ are completely determined by $\boldsymbol{A}, \boldsymbol{v}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{B}, \boldsymbol{t}, \boldsymbol{t}', u, \boldsymbol{y}$, the statistical distance between the distribution $(\boldsymbol{w}_1, \boldsymbol{w'}_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p, c, \boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_{s_3}, \boldsymbol{z}_p, \boldsymbol{z}_w, \boldsymbol{z}_B)$ generated by the simulation algorithm $S(\boldsymbol{A}, \boldsymbol{v}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{B})$ and the distribution of real protocol is within 2^{-100}.

Special Soundness: Let $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_{s_3}, \boldsymbol{z}_p, \boldsymbol{z}_w, \boldsymbol{z}_B, c)$ and $(\boldsymbol{z}^*, \boldsymbol{z}'^*, \boldsymbol{z}^*_{s_1}, \boldsymbol{z}^*_{s_2}, \boldsymbol{z}^*_{s_3}, \boldsymbol{z}^*_p, \boldsymbol{z}^*_w, \boldsymbol{z}^*_B, c^*)$ are two transcripts of real protocol with $c \neq c^*$. We are able to extract a valid opening $((\bar{\boldsymbol{z}}, \bar{\boldsymbol{z}}', \bar{\boldsymbol{z}}_{s_1}, \bar{\boldsymbol{z}}_{s_2}, \bar{\boldsymbol{z}}_{s_3}, \bar{\boldsymbol{z}}_p, \bar{\boldsymbol{z}}_w, \bar{\boldsymbol{z}}_B), \bar{i}, \bar{c})$ of commitments $\boldsymbol{t}, \boldsymbol{t}'$, where $\bar{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{z}^*$, $\bar{\boldsymbol{z}}' = \boldsymbol{z}' - \boldsymbol{z}'^*$, $\bar{\boldsymbol{z}}_{s_1} = \boldsymbol{z}_{s_1} - \boldsymbol{z}^*_{s_1}$, $\bar{\boldsymbol{z}}_{s_2} = \boldsymbol{z}_{s_2} - \boldsymbol{z}^*_{s_2}$, $\bar{\boldsymbol{z}}_{s_3} = \boldsymbol{z}_{s_3} - \boldsymbol{z}^*_{s_3}$, $\bar{\boldsymbol{z}}_p = \boldsymbol{z}_p - \boldsymbol{z}^*_p$, $\bar{\boldsymbol{z}}_w = \boldsymbol{z}_w - \boldsymbol{z}^*_w$, $\bar{\boldsymbol{z}}_B = \boldsymbol{z}_B - \boldsymbol{z}^*_B$, $\bar{c} = c - c^*$, $\bar{i} \in \mathbb{Z}_q$. Then the following equations hold:

$\bar{c}\boldsymbol{t} = Com(\bar{c}\bar{i}, \bar{\boldsymbol{z}}),\ \bar{c}\boldsymbol{t}' = Com(\bar{c}\bar{i}\delta, \bar{\boldsymbol{z}}')$,

$\bar{c}\boldsymbol{y} = \boldsymbol{B}\bar{\boldsymbol{z}}_B,\ \bar{c}u = \boldsymbol{v}^T\bar{\boldsymbol{z}}_s,\ \bar{c}\boldsymbol{M} = \boldsymbol{G}_1\bar{\boldsymbol{z}}_p + \boldsymbol{G}_2\bar{\boldsymbol{z}}_w$,

such that $||(\bar{\boldsymbol{z}}, \bar{\boldsymbol{z}}', \bar{\boldsymbol{z}}_B, \bar{\boldsymbol{z}}_p, \bar{\boldsymbol{z}}_w)|| \leq 2B_1$, $||\bar{\boldsymbol{z}}_{s_1}, \bar{\boldsymbol{z}}_{s_2}|| \leq 2B_2$, $||\bar{\boldsymbol{z}}_{s_3}|| \leq 2B_3$. This completes the proof.

The protocol $\prod_{\mathcal{PBGS}}$ is able to be converted into a NIZKPoK by Fiat-Shamir transformation. In order to do that, we define the hash function $H : \{0,1\}^* \to C$ that is used to generate challenge. And we let challenge $c = H(\boldsymbol{t}, \boldsymbol{t'}, \boldsymbol{v}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{y}, \delta, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{w}_1, \boldsymbol{w'}_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p, \boldsymbol{M})$. Then verifier $\mathcal{V}$ recovers $\boldsymbol{w}_1, \boldsymbol{w'}_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_B, \boldsymbol{w}_p$ from public information and obtains c'. If $c' = c$, $\mathcal{V}$ accepts the transcript and outputs 1; otherwise $\mathcal{V}$ returns 0.

4.2 PBGS Scheme

In this section, we show a scheme of PBGS from lattice specifically.

GSetup (1^λ):

Given a security parameter λ, the algorithm sets $d = O(\lambda)$ as a power of 2, a parameter $\ell > O(\log \lambda)$, integer bound $\beta = \text{poly}(d)$ and challenge bound $\kappa > 0$, prime modulus $q, Q \geq \beta\sqrt{d}$, Gaussian parameter $\sigma = q^{1/2} \cdot O(\sqrt{d})$. Set polynomial ring $R = \mathbb{Z}[X]/ < X^d + 1 >$, set of identity $[N] \subseteq \mathbb{Z}_q$, hash function $H : \{0,1\}^* \to C$. Let gadget matrix $\boldsymbol{g}^T = [\ 1 \quad \delta\ ] \in R_q^{1\times 2}$.

(a) Select $\boldsymbol{a}_1 = \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} \in R_q^3$, $\boldsymbol{a}_2 = \begin{bmatrix} 0 \\ 1 \\ a_3 \end{bmatrix} \in R_q^3$. Let $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix} \in R_q^{2\times 3}$.

(b) Select $\boldsymbol{a} \leftarrow R_q^2$, $\boldsymbol{R} \leftarrow \chi^{2\times 2}$. Let $\boldsymbol{b}^T = \boldsymbol{a}^T\boldsymbol{R} \in R_q^{1\times 2}$.

(c) Select $(\boldsymbol{s}_{0,1}, \boldsymbol{s}_{0,2}, \boldsymbol{s}_{0,3}) \leftarrow D_\sigma^2 \times D_\sigma^2 \times D_\sigma^3$. Set $u = \begin{bmatrix} \boldsymbol{a}^T & \boldsymbol{b}^T & \boldsymbol{a}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_{0,1} \\ \boldsymbol{s}_{0,2} \\ \boldsymbol{s}_{0,3} \end{bmatrix}$.

(d) Select $a \leftarrow R_Q$, $(\boldsymbol{s}, \boldsymbol{e}) \leftarrow \chi^3 \times \chi^3$. Set $\boldsymbol{b}_1 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = a\boldsymbol{s} + \boldsymbol{e} \in R_Q^3$.

(e) Select $\boldsymbol{G}_1 \leftarrow R_q^{1\times 3}$, $\boldsymbol{G}_2 \in R_q^{1\times(\ell-3)}$, where $\boldsymbol{G}_2$ is an approximate identity matrix. Define policy relation $\text{PR}_{G_1,G_2} : (\chi^3 \times R_q) \times \chi^{\ell-3} \to \{0,1\}$, where:

$$\text{PR}_{G_1,G_2}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) = 1 \Leftrightarrow \boldsymbol{G}_1 \cdot \boldsymbol{p} + \boldsymbol{G}_2 \cdot \boldsymbol{w} = \boldsymbol{M} \bmod q,$$

and $\boldsymbol{p} \leftarrow \chi^3$, message $\boldsymbol{M} \in R_q$, witness $\boldsymbol{w} \leftarrow \chi^{\ell-3}$.

(f) Output gpk $= (\boldsymbol{A}, a, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{b}_1, \boldsymbol{G}_1, \boldsymbol{G}_2, u)$, gmk $= \boldsymbol{R}$ and gtk $= \boldsymbol{s}$.

KeyGen ($\boldsymbol{R}$,$\boldsymbol{p}$, i):

Given group master private key $\boldsymbol{R}$, policy $\boldsymbol{p}$ and member $i \in [N]$, KGC will generate a signing key pair $\text{sk}_{\boldsymbol{p},i}$ in the following way:

(a) $(\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}) \leftarrow \text{SampleD}(\boldsymbol{a}_3, \boldsymbol{b}, \boldsymbol{R}, u - \boldsymbol{a}_2^T\boldsymbol{p}, \sigma)$ satisfying:

$$\left[\boldsymbol{a}^T | \boldsymbol{b}^T + i\boldsymbol{g}^T\right] \begin{bmatrix} \boldsymbol{s}_{i,1} \\ \boldsymbol{s}_{i,2} \end{bmatrix} = u - \boldsymbol{a}_2^T\boldsymbol{p}, \text{ where } (\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}) \in D_\sigma^2 \times D_\sigma^2.$$

(b) Output the signing key $\text{sk}_{\boldsymbol{p},i} = (\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2})$.

Sign ($\text{sk}_{\boldsymbol{p},i}, \boldsymbol{M}, \boldsymbol{w}$):

Given signing key $\text{sk}_{\boldsymbol{p},i}$, message $\boldsymbol{M} \in R_q^\ell$ and witness $\boldsymbol{w}$:

(a) If $\text{PR}_{G_1,G_2}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}) \neq 1$, then return $\perp$; otherwise perform the following steps.

(b) Select $(\boldsymbol{r}, \boldsymbol{r'}) \leftarrow \chi^3 \times \chi^3$. Set $\boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = Com(i, \boldsymbol{r})$, $\boldsymbol{t'} = \begin{bmatrix} t'_1 \\ t'_2 \end{bmatrix} = Com(i\delta, \boldsymbol{r'})$.

(c) Set $\boldsymbol{v}^T = \left[\boldsymbol{a}^T | \boldsymbol{b}^T + [t_2 | t'_2] | \boldsymbol{a}_2^T\right] \in R_q^{1\times 7}$, $\boldsymbol{s}' = \begin{bmatrix} \boldsymbol{s}_{i,1} \\ \boldsymbol{s}_{i,2} \\ \boldsymbol{p} - [\boldsymbol{r}|\boldsymbol{r}']\,\boldsymbol{s}_{i,2} \end{bmatrix} \in R_q^7$ satisfying $\boldsymbol{v}^T\boldsymbol{s}' = u$.

(d) Select $s_{\boldsymbol{B}} \leftarrow \chi$, $e_1 \leftarrow \chi$, $\boldsymbol{e}_2 \leftarrow \chi^3$, set $h = q(as_{\boldsymbol{B}} + e_1)$ and $\boldsymbol{d} = q(\boldsymbol{b}_1 s_{\boldsymbol{B}} + \boldsymbol{e}_2) + \boldsymbol{r}$.

(e) Set $\boldsymbol{B}_1 = \begin{bmatrix} qa & q & 0 & 0 & 0 & 0 & 0 & 0 \\ qb_1 & 0 & q & 0 & 0 & 1 & 0 & 0 \\ qb_2 & 0 & 0 & q & 0 & 0 & 1 & 0 \\ qb_3 & 0 & 0 & 0 & q & 0 & 0 & 1 \end{bmatrix} \in R_Q^{4\times 8}$, $\boldsymbol{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \boldsymbol{a}_1^T \end{bmatrix} \in R_q^{1\times 8}$ and $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \end{bmatrix}$.

(f) Set $\boldsymbol{r}_{\boldsymbol{B}} = \begin{bmatrix} s_{\boldsymbol{B}} \\ e_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{r} \end{bmatrix} \in R_q^8$ and $\boldsymbol{y} = \begin{bmatrix} h \\ \boldsymbol{d} \\ t_1 \end{bmatrix} \in R_Q^4 \times R_q$ satisfying $\boldsymbol{B}\boldsymbol{r}_{\boldsymbol{B}} = \boldsymbol{y}$.

(g) Generate a NIZKPoK $\prod = (\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{\boldsymbol{B}}, \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_{s_3}, \boldsymbol{z}_{\boldsymbol{p}}, \boldsymbol{z}_{\boldsymbol{w}}, c)$ to show the possession of $(\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, \boldsymbol{w}, \boldsymbol{r}_{\boldsymbol{B}})$ satisfying:

- $(\boldsymbol{p}, \boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2})$ is a valid signing key, and $\boldsymbol{v}^T\boldsymbol{s}' = u$.
- $\boldsymbol{G}_1 \cdot \boldsymbol{p} + \boldsymbol{G}_2 \cdot \boldsymbol{w} = \boldsymbol{M} \bmod q$.
- $(\boldsymbol{d}, h)$ is a valid verifiable ciphertext so that $\boldsymbol{B}\boldsymbol{r}_{\boldsymbol{B}} = \boldsymbol{y}$.

(h) Output the signature $\sum = (\boldsymbol{t}, \boldsymbol{t}', \prod, h, \boldsymbol{d})$.

Verify (gpk, $\sum$, $\boldsymbol{M}$):

Given gpk, signature $\sum$ and message $\boldsymbol{M}$:

(a) Recover $\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}, \boldsymbol{y}, \boldsymbol{v}$.

(b) Perform the verification in Section 4.1. If the verification algorithm accepts the $\prod$, output "Valid"; otherwise return "Invalid".

Open (gtk, $\sum$):

Given tracking key gtk and signature Σ:

(a) If the algorithm Verify returns "Invalid" for the signature Σ, output $\perp$ and terminate; otherwise perform the following steps.

(b) Select $c' \leftarrow C$, set $\bar{c} = c - c'$, where c is a challenge defined in Section 4.1.

(c) Set $\bar{\boldsymbol{r}} = (\boldsymbol{d} - h\boldsymbol{s})\bar{c} \bmod Q$. If $||\bar{\boldsymbol{r}}||_\infty \le Q/4\kappa$, $\bar{\boldsymbol{r}} = \bar{\boldsymbol{r}} \bmod q$; otherwise return $\perp$.

(d) Compute $i = t_2 - \boldsymbol{a}_2^T\bar{\boldsymbol{r}} \cdot \bar{c}^{-1}$. If $i \in [N]$, return i, otherwise return $\perp$.

5 Security Analysis

Theorem 2 (Correctness) The proposed PBGS scheme is correct with overwhelming probability.

Proof:

1) Verification correctness

For gpk, gmk, gtk $\leftarrow$ GSetup(1^λ), $\text{sk}_{p,i} \leftarrow$ KeyGen($\boldsymbol{R}, \boldsymbol{p}, i$), $\sum \leftarrow$ Sign($\text{sk}_{p,i}, \boldsymbol{M}, \boldsymbol{w}$), we compute $c' = H(\boldsymbol{t}, \boldsymbol{t}', \boldsymbol{v}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{y}, \delta, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{w}_1, \boldsymbol{w}'_1, \boldsymbol{w}_2, \boldsymbol{w}_s, \boldsymbol{w}_{\boldsymbol{B}}, \boldsymbol{w}_p, \boldsymbol{M})$ by the Verification equation in Protocol 1. Then $c' = c$ is hold with an overwhelming probability. Furthermore, the distribution $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{\boldsymbol{B}}, \boldsymbol{z}_{\boldsymbol{p}}, \boldsymbol{z}_{\boldsymbol{w}})$, $(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2})$, $\boldsymbol{z}_{s_3}$ is close to $D_{\xi_1}^{14+\ell}$, $D_{\xi_2}^4$, $D_{\xi_3}^3$ respectively after rejection sampling introduced in Lemma 4. And we have $||(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{\boldsymbol{B}}, \boldsymbol{z}_{\boldsymbol{p}}, \boldsymbol{z}_{\boldsymbol{w}})|| \le \sqrt{2(14+\ell)d}\xi_1$, $||(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2})|| \le 2\sqrt{2d}\xi_2$, $||\boldsymbol{z}_{s_3}|| \le \sqrt{6d}\xi_3$ according to Lemma 3. Therefore, the probability of "Invalid" $\leftarrow$ Verify(gpk, $\sum$, $\boldsymbol{M}$) is negligible.

2) Opening correctness

In signing phase, the signer generates verifiable ciphertext $(h, \boldsymbol{d})$ by encrypting the random $\boldsymbol{r}$. The ciphertext $(h, \boldsymbol{d})$ will be verified during the Verify phase. If the algorithm Verify returns "Valid", $(h, \boldsymbol{d})$ is a valid encryption about random r. Then administrator sets $\bar{c}$:

$$c' \leftarrow C,\ \bar{c} = c - c'.$$

And the following equation holds:

$$\begin{aligned}\bar{\boldsymbol{r}} &= (\boldsymbol{d} - h\boldsymbol{s})\bar{c} \bmod Q \\ &= p\bar{c}(\boldsymbol{b}_1 s_{\boldsymbol{B}} + \boldsymbol{e}_2) + \boldsymbol{r}\bar{c} - p\boldsymbol{s}\bar{c}(as_{\boldsymbol{B}} + e_1) \bmod Q \\ &= p\bar{c}(as_{\boldsymbol{B}}\boldsymbol{s} + s_{\boldsymbol{B}}\boldsymbol{e} + \boldsymbol{e}_2) - p\bar{c}(as_{\boldsymbol{B}}\boldsymbol{s} + e_1\boldsymbol{s}) + \boldsymbol{r}\bar{c} \bmod Q \\ &= p\bar{c}(s_{\boldsymbol{B}}\boldsymbol{e} + \boldsymbol{e}_2 - e_1\boldsymbol{s}) + \boldsymbol{r}\bar{c} \bmod Q.\end{aligned}$$

According to [26], we know that $\|p\bar{c}(s_{\boldsymbol{B}}\boldsymbol{e} + \boldsymbol{e}_2 - e_1\boldsymbol{s}) + \boldsymbol{r}\bar{c}\|_\infty \le \|p(\bar{s}_{\boldsymbol{B}}\boldsymbol{e} + \bar{\boldsymbol{e}}_2 - \bar{e}_1\boldsymbol{s}) + \boldsymbol{r}\|_\infty \le Q/4\kappa$, which is $\bar{\boldsymbol{r}} \le Q/4\kappa$. And administrator computes $\boldsymbol{r}\bar{c} = \bar{\boldsymbol{r}} \bmod q$ to open the commitment:

$$\boldsymbol{a}_1^T\bar{\boldsymbol{r}} = \bar{c}t_1 \bmod q,\ \boldsymbol{a}_2^T\bar{\boldsymbol{r}} + \bar{c}i = \bar{c}t_2 \bmod q.$$

From which we get $i = t_2 - \boldsymbol{a}_2^T\bar{\boldsymbol{r}} \cdot \bar{c}^{-1}$. Therefore, the probability of $\perp \leftarrow \mathrm{Open}(\mathrm{gtk}, \sum)$ is negligible.

Theorem 3 (Simulatability) The proposed PBGS scheme meets simulatability defined in Definition 6 under ROM, if the $\mathrm{MLWE}_{1,\chi}$ problem is hard.

Proof: We will construct a PPT algorithm SimSign, which returns a simulated signature $\sum^*$ by inputting arbitrary message $\boldsymbol{M} \in R_q$. Specifically, the SimSign algorithm is similar to honest signature algorithm roughly, except for the following modifications:

1) For commitments $\boldsymbol{t}$ and $\boldsymbol{t}'$, we modify the $(i, \boldsymbol{r})$ as a random $(i^*, \boldsymbol{r}^*)$. Due to the hiding of commitment in Lemma 6, the algorithm SimSign is still indistinguishable from the honest signature algorithm.
2) For NIZKPoK $\prod$, we modify the $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \boldsymbol{z}_p, \boldsymbol{z}_w)$, $\boldsymbol{z}_{\boldsymbol{B}}$, $\boldsymbol{z}_{s_3}$ as random values selected from $D_{\xi_1}^{14+\ell}$, $D_{\xi_2}^4$, $D_{\xi_3}^3$ according to the simulation algorithm of Theorem 1, and get $\prod^* = (\boldsymbol{z}^*, \boldsymbol{z}'^*, \boldsymbol{z}_{\boldsymbol{B}}^*, \boldsymbol{z}_{s_1}^*, \boldsymbol{z}_{s_2}^*, \boldsymbol{z}_{s_3}^*, \boldsymbol{z}_p^*, \boldsymbol{z}_w^*, c)$. Then the statistical distance between $\prod$ and $\prod^*$ is within 2^{-100}.
3) For ciphertext $(h, \boldsymbol{d})$, we set $h^* = qa$ and $\boldsymbol{d}^* = q\boldsymbol{b}_1$. Then the $(h^*, \boldsymbol{d}^*)$ is indistinguishable from $(h, \boldsymbol{d})$ under the $\mathrm{MLWE}_{1,\chi}$ problem.

As a result, the algorithm SimSign is able to generate a simulated signature $\sum^* = (\boldsymbol{t}^*, \boldsymbol{t}'^*, \prod^*, h^*, \boldsymbol{d}^*)$, which is indistinguishable from the legitimate signature generated by the honest signature algorithm. And the SimSign does not need any signing key or policy.

After obtaining the algorithm SimSign, challenger $\mathcal{C}$ runs the GSetup (1^λ) honestly and sends the gpk and gmk to attacker $\mathcal{A}$. $\mathcal{A}$ adaptively chooses policies $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_Q$ and queries signing key of $\boldsymbol{p}_i$. $\mathcal{C}$ runs $\mathrm{sk}_{p,i} \leftarrow \mathrm{KeyGen}(\mathrm{gmk}, \boldsymbol{p}, i)$ and sends $\mathrm{sk}_{p,i}$ to $\mathcal{A}$. Next $\mathcal{A}$ chooses $i \in [Q]$, $\boldsymbol{M} \in R_q$, $\boldsymbol{p} \leftarrow \chi^3$, $\boldsymbol{w}^* \leftarrow \chi^{\ell-3}$ and sends them to $\mathcal{C}$. If $\mathrm{PR}((\boldsymbol{p}, \boldsymbol{M}), \boldsymbol{w}^*) = 0$, the game will be terminated; otherwise $\mathcal{C}$ computes simulated signature $\sum_0 \leftarrow \mathrm{SimSign}(\boldsymbol{M}^*)$ and legitimate signature $\sum_1 \leftarrow \mathrm{Sign}(\mathrm{sk}_{p,i}, \boldsymbol{M}^*, \boldsymbol{w}^*)$. Finally, $\mathcal{C}$ selects a bit $b \in \{0, 1\}$ and sends $\sum_b$ to $\mathcal{A}$.

Since the simulated signature $\sum_0$ is indistinguishable from the legitimate signature $\sum_1$, the probability that $\mathcal{A}$ correctly guess the bit b is $1/2 + \text{negl}(n)$. That is, the advantage of $\mathcal{A}$ breaking the simulatability of our PBGS scheme is negligible.

Theorem 4 (Full Traceability) The proposed PBGS scheme meets full traceability defined in Definition 7 under ROM, if the $\text{MSIS}_{5,\beta}$ problem is hard.

Proof: Assume that an attacker $\mathcal{A}$ successfully forges an untraceable signature with non-negligible probability ε. Then a challenger $\mathcal{C}$ will construct a non-zero solution about the MSIS problem by the result of $\mathcal{A}$ with non-negligible probability. Specifically, $\mathcal{C}$ initializes the list Γ, I and runs the GSetup (1^λ) honestly. The gpk and gtk are sent to $\mathcal{A}$. Next $\mathcal{C}$ selects $j \in [N]$, $\boldsymbol{p}_j$. $\mathcal{A}$ have access to the queries of signing key and signature defined in Definition 7.

Finally, $\mathcal{A}$ outputs a signature $\sum = (\boldsymbol{t}, \boldsymbol{t}', \prod, h, \boldsymbol{d})$ about message $\boldsymbol{M}^* \in R_q$, which satisfies "Valid" $\leftarrow$ Verify(gpk, $\sum, \boldsymbol{M}^*$), and $\perp \leftarrow$ Open(gtk, $\sum$) or $j \leftarrow$ Open(gtk, $\sum$), where $\{\boldsymbol{p}_j\} \notin \Gamma$ and $(\boldsymbol{M}^*, \sum) \notin I$. According to the special soundness of Theorem 1, there are two different challenges $\mathcal{C}$ can extract $\bar{\boldsymbol{z}}, \bar{\boldsymbol{z}}' \in R^3$, $\bar{i} \in \mathbb{Z}_q$, $\bar{\boldsymbol{z}}_{s_1}, \bar{\boldsymbol{z}}_{s_2} \in R^2$, $\bar{\boldsymbol{z}}_{s_3} \in R^3$, $\bar{\boldsymbol{z}}_{\boldsymbol{B}} \in R^8$, $\bar{\boldsymbol{z}}_{\boldsymbol{p}} \in R^3$, $\bar{\boldsymbol{z}}_w \in R^{\ell-3}$ $\bar{c} \in \bar{C}$ satisfying $||(\bar{\boldsymbol{z}}, \bar{\boldsymbol{z}}', \bar{\boldsymbol{z}}_{\boldsymbol{B}}, \bar{\boldsymbol{z}}_{\boldsymbol{p}}, \bar{\boldsymbol{z}}_w)|| \le 2B_1$, $||\bar{\boldsymbol{z}}_{s_1}, \bar{\boldsymbol{z}}_{s_2}|| \le 2B_2$, $||\bar{\boldsymbol{z}}_{s_3}|| \le 2B_3$. We will get that the probability of completing the above extraction of $\mathcal{C}$ is at least $\varepsilon(\frac{\varepsilon}{h_1} - 2^{-\lambda})$ by the forking lemma of [32], where $h_1 \ge 2$ is the length of the hash function H. For ciphertext $(h, \boldsymbol{d})$, $\mathcal{C}$ will decrypt and obtain $(\tilde{\boldsymbol{r}}, \tilde{c})$ by the tracking key gtk. According to the soundness of the verifiable encryption scheme from [26], we know that $\tilde{\boldsymbol{r}}\bar{c} = \bar{\boldsymbol{z}}\tilde{c}$ will hold with overwhelming probability, which means that Open(gtk, $\sum$) $\in \mathbb{Z}_q$. Therefore, the probability of $\perp \leftarrow$ Open(gtk, $\sum$) is negligible. Since the set of identity $[N]$ is a uniform distribution, the probability of $i = j$ in forged signature is $1/N$. Assuming that $i = j$, then:

$$\bar{c}t_2 = \boldsymbol{a}_2^T\bar{\boldsymbol{z}} + \bar{c}j,\ \bar{c}t_2{}' = \boldsymbol{a}_2^T\bar{\boldsymbol{z}}' + \bar{c}j\delta,\ \left[\boldsymbol{a}^T|\boldsymbol{b}^T + [t_2|t_2'] - j\boldsymbol{g}^T|\boldsymbol{a}_2^T\right]\bar{\boldsymbol{z}}_s = \bar{c}u.$$

Multiplying by $\bar{c}$ and replacing $[t_2|t_2']\bar{c}$:

$$\left[\bar{c}\boldsymbol{a}^T|\bar{c}\boldsymbol{b}^T + \left[\boldsymbol{a}_2^T\bar{\boldsymbol{z}}|\boldsymbol{a}_2^T\bar{\boldsymbol{z}}'\right]|\bar{c}\boldsymbol{a}_2^T\right]\bar{\boldsymbol{z}}_s = \bar{c}^2u.$$

$$\tilde{\boldsymbol{z}} = \begin{bmatrix} \bar{\boldsymbol{z}}_1\bar{c} + \boldsymbol{R}\bar{\boldsymbol{z}}_2\bar{c} \\ \bar{\boldsymbol{z}}_3\bar{c} + \left[\bar{\boldsymbol{z}} \quad \bar{\boldsymbol{z}}'\right]\bar{\boldsymbol{z}}_2 \end{bmatrix},$$

where $\boldsymbol{R}$ is master private key. Then:

$$\left[\boldsymbol{a}^T|\boldsymbol{a}_2^T\right]\tilde{\boldsymbol{z}} = \bar{c}^2u.$$

Then $\mathcal{C}$ performs algorithm Sample D by $\boldsymbol{R}$ to obtain $\boldsymbol{s}_j$, which fulfills $\left[\boldsymbol{a}^T|\boldsymbol{b}|\boldsymbol{a}_2^T\right]\boldsymbol{s}_j = u$ and is unknown to $\mathcal{A}$ in forgery phase. Let $\boldsymbol{s}_j^* = \begin{bmatrix} \boldsymbol{s}_{j_1} + \boldsymbol{R}\boldsymbol{s}_{j_2} \\ \boldsymbol{s}_{j_3} \end{bmatrix}$, we obtain $\left[\boldsymbol{a}^T|\boldsymbol{a}_2^T\right]\boldsymbol{s}_j^* = u$, where the probability of $\bar{c}\boldsymbol{s}_j^* = \tilde{\boldsymbol{z}}$ is negligible. Then $\mathcal{C}$ has constructed the equation:

$$\left[\boldsymbol{a}^T|\boldsymbol{a}_2^T\right] \cdot (\tilde{\boldsymbol{z}} - \bar{c}^2\boldsymbol{s}_j^*) = \boldsymbol{0}.$$

And the bound on the norm of the solution satisfies:

$$\begin{aligned}\left\|\tilde{\boldsymbol{z}}-\bar{c}^2\boldsymbol{s}_j^*\right\| &\leq \|\tilde{\boldsymbol{z}}\|+4\kappa^2\left\|\boldsymbol{s}_j^*\right\| \\ &\leq 2\kappa\|\bar{\boldsymbol{z}}_1\|+4\kappa\sqrt{d}\|\bar{\boldsymbol{z}}_2\|+2\kappa\|\bar{\boldsymbol{z}}_3\|+\sqrt{6d}\|\bar{\boldsymbol{z}}_2\| \\ &+4\kappa^2(1+3\sqrt{d})(2\sqrt{d}\sigma)+4\kappa^2\sqrt{6d}\sigma \\ &\ll q.\end{aligned}$$

Hence, $\mathcal{C}$ constructs a solution of $\mathrm{MSIS}_{5,\beta}$ problem with a probability of $\varepsilon\cdot\frac{1}{N}\cdot\left(\frac{\varepsilon}{h_1}-2^{-\lambda}\right)$. Since the probability of successful forgery by attacker $\mathcal{A}$ is non-negligible, the probability of $\varepsilon\cdot\frac{1}{N}\cdot\left(\frac{\varepsilon}{h_1}-2^{-\lambda}\right)$ is also non-negligible.

6 Efficiency Analysis

In this section, we choose three schemes of GS from lattice to carry out efficiency analysis and comparison with our scheme. We will perform a detailed analysis of the storage overhead of group public key, administrator tracking key, members signing key and signature. Firstly, we fix the security parameter λ and the maximum number of members N. Other parameters will be set as described in Section 4.2. Specifically, we set $N=2^{12}$, $\ell=4$, $\kappa=26$, dimension $d=2^{12}$, Gaussian parameter $\sigma=6\sqrt{dq}\approx 1.01\cdot 10^8$, modulus q and Q are 2^{36}, 2^{72} respectively. Then we get a root-hermite factor by definition from [33]. Such a factor means that the parameters we chose guarantee $\lambda=93$ bits space security against quantum adversaries. The comparison for the storage cost of the GS is listed in Tab. 2.

Table 2: Comparison of storage overhead for security level $\lambda=93$ bits

Scheme	Group public key	Tracking key	Signing key	Signature
[17]	8194.0 KB	40.0 KB	38.6 KB	234.0 KB
[11]	488.0 KB	40.0 KB	252.0 KB	1053.0 KB
[16]	513.0 KB	88.3 KB	123.0 KB	931.8 KB
Ours	126.5 KB	16.5 KB	38.5 KB	209.0 KB

Compared with the above three schemes of GS, our construction has lower storage overhead on key and signature to a certain extent. The size of key decreased roughly by 83.13% and the size of signature is also decreased roughly by 46.01%.

Funding Statement: This work is supported by the National Natural Science Foundation of China (61802117), Support Plan of Scientific and Technological Innovation Team in Universities of Henan Province (20IRTSTHN013), the Youth Backbone Teacher Support Program of Henan Polytechnic University under Grant (2018XQG-10).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Bellare and G. Fuchsbauer, "Policy-based signatures," in *Public-Key Cryptography, PKC 2014. Proc.: Lecture Notes in Computer Science (LNCS 8383)*, Berlin, Germany, pp. 520–537, 2014.

[2] S. Cheng, K. Nguyen and H. Wang, "Policy-based signature scheme from lattices," *Designs, Codes and Cryptography*, vol. 81, no. 1, pp. 43–74, 2016.

[3] D. Chaum and E. V. Heyst, "Group signatures," in *EUROCRYPT 1991. Proc.: Lecture Notes in Computer Science (LNCS 547)*, Berlin, Germany, pp. 257–265, 1991.

[4] S. D. Gordon, J. Katz and V. Vaikuntanathan, "A group signature scheme from lattice assumptions," in *ASIACRYPT 2010. Proc.: Lecture Notes in Computer Science (LNCS 6477)*, Berlin, Germany, pp. 395–412, 2010.

[5] F. Laguillaumie, A. Langlois, B. Libert and D. Stehlé, "Lattice-based group signatures with logarithmic signature size," in *ASIACRYPT 2013. Proc.: Lecture Notes in Computer Science (LNCS 8270)*, Berlin, Germany, pp. 41–61, 2013.

[6] S. Ling, K. Nguyen and H. Wang, "Group signatures from lattices: simpler, tighter, shorter, ring-based," in *Public-Key Cryptography, PKC 2015. Proc.: Lecture Notes in Computer Science (LNCS 9020)*, Berlin, Germany, pp. 427–449, 2015.

[7] B. Libert, S. Ling, F. Mouhartem, K. Nguyen and H. Wang, "Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions," in *ASIACRYPT 2016. Proc.: Lecture Notes in Computer Science (LNCS 10032)*, Berlin, Germany, pp. 373–403, 2016.

[8] B. Libert, S. Ling, K. Nguyen and H. Wang, "Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors," in *EUROCRYPT 2016. Proc.: Lecture Notes in Computer Science (LNCS 9666)*, Berlin, Germany, pp. 1–31, 2016.

[9] B. Libert, F. Mouhartem and K. Nguyen, "A Lattice-based group signature scheme with message-dependent opening," in *Applied Cryptography and Network Security, ACNS 2016. Proc.: Lecture Notes in Computer Science (LNCS 9696)*, Cham, Switzerland, pp. 137–155, 2016.

[10] S. Ling, K. Nguyen, H. Wang and Y. Xu, "Lattice-based group signatures: achieving full dynamicity with ease," in *Applied Cryptography and Network Security, ACNS 2017. Proc.: Lecture Notes in Computer Science (LNCS 10355)*, Cham, Switzerland, pp. 293–312, 2017.

[11] S. Ling, K. Nguyen, H. Wang and Y. Xu, "Constant-size group signatures from lattices," in *Public-Key Cryptography, PKC 2018. Proc.: Lecture Notes in Computer Science (LNCS 10770)*, Cham, Switzerland, pp. 58–88, 2018.

[12] Y. Zhang, X. Liu, Y. Yin, Q. Zhang and H. Jia, "On new zero-knowledge proofs for fully anonymous lattice-based group signature scheme with verifier-local revocation," in *Applied Cryptography and Network Security, ACNS 2020. Proc.: Lecture Notes in Computer Science (LNCS 12418)*, Cham, Switzerland, pp. 381–399, 2020.

[13] L. Ducas and D. Micciancio, "Improved short lattice signatures in the standard model," in *CRYPTO 2014. Proc.: Lecture Notes in Computer Science (LNCS 8616)*, Berlin, Germany, pp. 335–352, 2014.

[14] R. Pino, V. Lyubashevsky and G. Seiler, "Lattice-based group signatures and zero-knowledge proofs of automorphism stability," in *Proc. ACM SIGSAC Conf. on Computer and Communications Security*, New York City, NY, USA, pp. 574–591, 2018.

[15] S. Agrawal, D. Boneh and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *CRYPTO 2010. Proc.: Lecture Notes in Computer Science (LNCS 6223)*, Berlin, Germany, pp. 98–115, 2010.

[16] C. Boschini, J. Camenisch and G. Neven, "Floppy-sized group signatures from lattices," in *Applied Cryptography and Network Security, ACNS 2018. Proc.: Lecture Notes in Computer Science (LNCS 10892)*, Cham, Switzerland, pp. 163–182, 2018.

[17] S. Katsumata and S. Yamada, "Group signatures without NIZK: from lattices in the standard model," in *EUROCRYPT 2019. Proc.: Lecture Notes in Computer Science (LNCS 11478)*, Cham, Switzerland, pp. 312–344, 2019.

[18] Y. Sun and Y. Liu, "A Lattice-based fully dynamic group signature scheme without NIZK," in *Information Security and Cryptology, Inscrypt 2020. Proc.: Lecture Notes in Computer Science (LNCS 12612)*, Cham, Switzerland, pp. 359–367, 2020.
[19] S. Canard, A. Georgescu, G. Kaim, A. R. Langlois and J. Traoré, "Constant-size lattice-based group signature with forward security in the standard model," in *Provable and Practical Security, ProvSec 2020. Proc.: Lecture Notes in Computer Science (LNCS 12505)*, Cham, Switzerland, pp. 24–44, 2020.
[20] M. Bellare, D. Micciancio and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *EUROCRYPT 2003. Proc.: Lecture Notes in Computer Science (LNCS 2656)*, Berlin, Germany, pp. 614–629, 2003.
[21] S. Doss, J. Paranthaman, S. Gopalakrishnan, A. Duraisamy, S. Pal *et al.,* "Memetic optimization with cryptographic encryption for secure medical data transmission in IoT-based distributed systems," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1577–1594, 2021.
[22] X. Zhang, X. Sun, X. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
[23] X. Zhang, W. Zhang, W. Sun, X. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
[24] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *EUROCRYPT 2012. Proc.: Lecture Notes in Computer Science (LNCS 7237)*, Berlin, Germany, pp. 700–718, 2012.
[25] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner and C. Peikert, "More efficient commitments from structured lattice assumptions," in *Security and Cryptography for Networks, SCN 2018. Proc.: Lecture Notes in Computer Science (LNCS 11035)*, Cham, Switzerland, pp. 614–629, 2018.
[26] V. Lyubashevsky and G. Neven, "One-shot verifiable encryption from lattices," in *EUROCRYPT 2017. Proc.: Lecture Notes in Computer Science (LNCS 10210)*, Cham, Switzerland, pp. 293–323, 2017.
[27] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 1, pp. 565–599, 2015.
[28] V. Lyubashevsky, K. Nguyen and G. Seiler, "Practical lattice-based zero-knowledge proofs for integer relations," in *Proc. ACM SIGSAC Conf. on Computer and Communications Security*, New York City, NY, USA, pp. 1051–1070, 2020.
[29] V. Lyubashevsky, "Lattice signatures without trapdoors," in *EUROCRYPT 2012. Proc.: Lecture Notes in Computer Science (LNCS 7237)*, Berlin, Germany, pp. 738–755, 2012.
[30] G. Xu, Y. Cao, S. Xu, K. Xiao, X. Liu *et al.,* "A novel post-quantum blind signature for log system in blockchain," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 945–958, 2022.
[31] L. Mei, C. Xu, L. Xu, X. Yu and C. Zuo, "Verifiable identity-based encryption with keyword search for IoT from lattice," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 2299–2314, 2021.
[32] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. 13th ACM Conf. on Computer and Communications Security*, New York City, NY, USA, pp. 390–399, 2006.
[33] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *EUROCRYPT 2008. Proc.: Lecture Notes in Computer Science (LNCS 4965)*, Berlin, Germany, pp. 31–51, 2008.