International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 15, Number 2, pages 153–181, July 2025

Efficient Group Signatures with Designated Traceability over Openers' Attributes from Lattices

Hiroaki Anada
Department of Mathematical Informatics, Faculty of Mathematical Informatics, Meiji Gakuin University
1518 Kamikurata-cho, Totsuka-ku, Yokohama-shi, Kanagawa, 244-8539 Japan
hiroaki.anada@mi.meijigakuin.ac.jp

Masayuki Fukumitsu
Department of Information Security, Faculty of Information Systems, University of Nagasaki
1-1-1 Manabino, Nagayo-cho, Nishisonogi-gun, Nagasaki, 851-2195 Japan
fukumitsu@sun.ac.jp

Shingo Hasegawa
Faculty of Symbiotic Systems Science, Fukushima University
1 Kanayagawa, Fukushima-shi, Fukushima, 960-1296 Japan
hasegawa@sss.fukushima-u.ac.jp

Received: February 14, 2025
Revised: May 2, 2025
Accepted: May 29, 2025
Communicated by Toru Nakanishi

Abstract

The *group signature with designated traceability* (GSdT) is a kind of group signatures (GS) which aim to restrict the opening authority of the group manager; by setting an access structure over openers' attributes at the signing, a signer is able to control openers who can open the signature. A generic construction of GSdT was given when the notion was introduced, then a pairing-based construction and a symmetric-key-based one were presented. Nonetheless, it remains open whether a post-quantum GSdT with full anonymity can be truly constructed.

In this paper, we give a lattice-based GSdT scheme that has full anonymity for the first time. In our construction, the lattice-based ciphertext-policy attribute-based encryption (CP-ABE) by Tsabary and the lattice-based group signatures (GS) by Libert et al. are employed. The CP-ABE is based on the Regev public-key encryption, while the GS uses a non-interactive zero-knowledge proof to prove the correctness of the encryption in the signing process. Based on the compatibility, we combine and modify them to build up a GSdT scheme.

Keywords: group signature, openers, attributes, lattices

1 Introduction

1.1 Background

The notion of group signature (GS) was proposed by Chaum and van Heyst [14]. The GS enables group members to sign a message on behalf of the group. There are two representative properties on

GS, called the anonymity and the traceability. The anonymity guarantees hide the signer's identity; that is, a signature does not reveal the actual signer in the group. The traceability allows an entity called an opener, by using a trapdoor key, to identify the actual signer from the signature.

In a GS scheme, the opener can open all signatures and know each actual signer. This means that the opener has excessive authority in the system. To resolve or mitigate this problem, several approaches have been taken to restrict the opening ability appropriately. The group signature with message-dependent opening [28] partially separates the opening functionality, which used to belong to only the opener, into a new entity called an admitter. In the opening process, the admitter issues a message-specific token that allows the opener to open a signature of the designated message. Then, the opener opens the signature corresponding to the message to identify the actual signer by using his secret opening key and the message-specific token. The notion of accountable tracing [25],[19] aims to divide the group into two kinds of users: the one consists of users who can be traced and the other consists of users who cannot be traced. Which one a user will be is determined at the time that a user joins the group. However, the signers themselves cannot control the right to the tracing. The bifurcated anonymous signatures [24] enable signers to choose whether a signature is traceable or not at the signing. On the other hand, the opener no longer has the right to trace when the signer generates an untraceable signature.

As one of the directions, we pay attention to the accountable ring signature that was initiated by Xu and Yung [30]. The accountable ring signature allows the signer (in an ad hoc ring of signers) to choose and determine an opener who can trace the signer. In other words, accountable ring signatures are fully anonymous for all undesignated openers, whereas only the designated opener can know who is the actual signer. Therefore, these signatures aim to guarantee the compatibility of the rights between the anonymity and the traceability by involving the signer in the tracing functionality. However, there is a risk that the opener will be revoked, and the designating signatures cannot be opened.

To realize a flexible option of signers over openers, the group signature with designated traceability (GSdT) was recently introduced [2, 3]. In GSdT, a signer can control a set of openers who can open the signature by setting an *access structure over openers' attributes* at the signing. GSdT has an advantage over the accountable ring signature in the sense that a signer can designate multiple openers via the access structure. Besides, the tracing functionality is maintained even when all the openers are revoked because new openers with satisfying attributes can be added. In [2], [3], the notion of GSdT is proposed and the generic construction is given from a ciphertext-policy attribute-based encryption (CP-ABE), a digital signature and a non-interactive zero-knowledge proof (NIZK). For specific constructions, a pairing-based construction [4] was given, and a symmetric-key-based one [5] was proposed. Due to the symmetric-key primitives, the latter scheme is expected to be secure against the computational power of quantum computers. However, the anonymity achieved in [5] is weaker than the original definition of the anonymity of the GSdT [3]. In this sense, there exists no GSdT yet that achieves both quantum-resistance and full anonymity.

1.2 Contributions

In this paper, we introduce the first lattice-based GSdT scheme that has full anonymity. Although there exists a generic construction of GSdT from [2, 3], we take a different approach on the construction. The generic construction of [2, 3] is in the sign-then-encrypt-then-prove paradigm like the construction of the ordinary group signature [9]. We find that some lattice-based group signatures such as [12, 20, 25, 10] are in the encrypt-then-prove paradigm, which are simpler constructions than the generic construction of group signature. These constructions in the latter paradigm take advantage of the compatibility of building blocks used. Thus, it is natural to explore constructions in the latter paradigm.

We employ the lattice-based CP-ABE by Tsabary [29] and the lattice-based GS by [22] (LLMNW GS) in our construction. These two schemes have a good feature in common when they are combined. That is, they use the Regev public-key encryption (Regev PKE) [26]. Tsabary's CP-ABE extends the Regev PKE [26] into the CP-ABE case. On the other hand, the LLMNW GS uses the NIZK proof which proves the correctness of the encryption by the Regev PKE. We can capture this feature

in our GSdT construction as in the construction of group signatures in the encrypt-then-prove paradigm. Our result is not only the first lattice-based GSdT scheme but also gives a new technique for constructing a GSdT scheme. Since our construction is specific to the building blocks used, it remains open whether or not we can generalize our construction for other cases.

We also compare the asymptotic efficiency of our proposed scheme with the lattice-based construction, which is yielded by applying the generic construction [3] to the Tsabary CP-ABE and the pair of the signature and the NIZK which is the same as LLMNW GS, and the GSdT from symmetric-key primitives [5]. As a result, we can find that the sizes of a group public key, an opening key, a group secret key and a group signature of ours are significantly shorter than those of [3]. Moreover, the computational times of joining, signing, opening and judging are asymptotically more efficient than theirs. On the other hand, ours realizes a post-quantum construction in the (partially) dynamic model [9] as the original syntax by [3] for the class of access structures richer than [5]. Moreover, the sizes of keys for ours are independent of the size of the attribute universe.

We finally note one limitation of our construction. Access structures supported in our GSdT are only conjunctive normal forms whose clauses have t bits of input (t-CNF). Constructing lattice-based GSdT with richer access structures is another interesting open question.

1.3 Difference from Our Conference Proceeding [6]

We give full proofs for all the security requirements of our proposed GSdT. We also give all syntax and security definitions of sub-algorithms. Note that we slightly revise the definition of the anonymity from [6] so that we eliminate an opener who can open the target group signature. This modification seems natural since such an opener trivially breaks the anonymity. Finally, we revise the asymptotic evaluation of our proposed GSdT. In the proceeding version, we have evaluated the efficiency by using all the parameters displayed in Table 1. On the other hand, we reevaluate them based solely on the security parameter λ and the parameters independent of λ such as the number of group members, the lengths of attributes and access structures. We also add the comparison with the GSdT from symmetric-key primitives [5].

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$ be the sets of all natural numbers, integers and real numbers, respectively. For any integers $a \leq b$, $[a, b] \subseteq \mathbb{Z}$ stands for the set of all integers x satisfying $a \leq x \leq b$. For any natural number a, $[\pm a]$ denotes the set $[-a, a]$. For any $a \in \mathbb{Z}$ and any positive odd number N, $a \bmod^{\pm} N$ means that $x = a \bmod N$ such that its representation is in the range $[\pm(N-1)/2]$.

Let $\boldsymbol{b} \in \mathbb{Z}^n$ be a vector with n dimensions. Suppose that $\boldsymbol{b} \in \mathbb{Z}^n$ is a row vector. We write $\|\boldsymbol{b}\|_2$ and $\|\boldsymbol{b}\|$ to denote the Euclidean norm and the infinite norm of $\boldsymbol{b}$, respectively. $\boldsymbol{b}^T$ is the transpose of $\boldsymbol{b}$. $\mathsf{bin}(\boldsymbol{b})$ stands for the binary representation of $\boldsymbol{b}$. For any $a \in \mathbb{N}$, $\boldsymbol{a}^n$ means $\begin{bmatrix} a & a & \cdots & a \end{bmatrix}^T \in \mathbb{Z}^n$. Any string $s \in \{0,1\}^n$ can be decomposed into $s[1], \ldots, s[n]$, where $s[i]$ is the i-th bit of s.

For any vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^n$, we denote by $[\boldsymbol{a}|\boldsymbol{b}] \in \mathbb{Z}^{n \times 2}$ the concatenation of the columns of $\boldsymbol{a}$ and $\boldsymbol{b}$. $\left[\frac{\boldsymbol{a}}{\boldsymbol{b}}\right] \in \mathbb{Z}^{2n}$ means the concatenation of the rows of $\boldsymbol{a}$ and $\boldsymbol{b}$. The similar notations are also used for matrices. For any full column-rank matrix $\boldsymbol{B} \in \mathbb{R}^{n \times m}$, $\tilde{\boldsymbol{B}}$ stands for the Gram-Schmidt orthogonalization.

For any distribution $\mathcal{D}$ over a set X, we write $x \leftarrow\!\!\$\ \mathcal{D}$ to denote that $x \in X$ is sampled according to $\mathcal{D}$. In particular, we simply represent $x \leftarrow\!\!\$\ X$ when $\mathcal{D}$ is the uniform distribution over a finite set X. For any real number $\epsilon \geq 0$, and any parameterized ensembles $(\mathcal{D}_{1,\lambda})_{\lambda \in \mathbb{N}}$ and $(\mathcal{D}_{2,\lambda})_{\lambda \in \mathbb{N}}$ of distributions over sets X_λ, we say that $(\mathcal{D}_{1,\lambda})_{\lambda \in \mathbb{N}}$ is *ϵ-close to* $(\mathcal{D}_{2,\lambda})_{\lambda \in \mathbb{N}}$ if the statistical distance $(1/2)\sum_{x \in X_\lambda} |\mathcal{D}_{1,\lambda}(x) - \mathcal{D}_{2,\lambda}(x)| = \epsilon(\lambda)$ for sufficiently large λ. A function ϵ is said to be *negligible in λ* if for any polynomial p, there exists $\lambda_0 \in \mathbb{N}$ such that $\epsilon(\lambda) < 1/p(\lambda)$ for any $\lambda \geq \lambda_0$. When $(\mathcal{D}_{1,\lambda})_{\lambda \in \mathbb{N}}$ is ϵ-close to $(\mathcal{D}_{2,\lambda})_{\lambda \in \mathbb{N}}$ for some negligible function ϵ, $(\mathcal{D}_{1,\lambda})_{\lambda \in \mathbb{N}}$ is *statistically close to* $(\mathcal{D}_{2,\lambda})_{\lambda \in \mathbb{N}}$. "probabilistic polynomial time" and "deterministic polynomial time" are abbreviated to PPT and DPT, respectively.

2.1 Lattices

Let $m \geq n \geq 1$, and let q be a prime. A *lattice* $\mathcal{L}$ in $\mathbb{R}^n$ with basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{Z}^n$ is defined by all integer linear combinations of the m basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$. For any matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and any vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, we set the following:

$$\begin{aligned}
\Lambda_q(\boldsymbol{A}) &= \left\{\boldsymbol{e} \in \mathbb{Z}^m \mid \exists \boldsymbol{s} \in \mathbb{Z}_q^n \text{ s.t. } \boldsymbol{A}^T\boldsymbol{s} = \boldsymbol{e} \bmod q\right\}, \\
\Lambda_q^{\perp}(\boldsymbol{A}) &= \{\boldsymbol{e} \in \mathbb{Z}^m \mid \boldsymbol{A}\boldsymbol{e} = \boldsymbol{0} \bmod q\}, \text{ and} \\
\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) &= \{\boldsymbol{e} \in \mathbb{Z}^m \mid \boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \bmod q\}.
\end{aligned}$$

Let $\mathcal{L} \subseteq \mathbb{R}^m$ be a lattice, let $\boldsymbol{c} \in \mathbb{R}^n$ and let $\sigma > 0$ be a real number. We set $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x} - \boldsymbol{c}\|^2/\sigma^2)$. Then, the discrete Gaussian distribution $D_{\mathcal{L},\sigma,\boldsymbol{c}}$ of support $\mathcal{L}$, standard deviation σ and center $\boldsymbol{c}$ is $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{y})/\rho_{\sigma,\boldsymbol{c}}(\mathcal{L})$ for any $\boldsymbol{y} \in \mathcal{L}$. We simply represent $D_{\mathcal{L},\sigma}$ when $\boldsymbol{c} = \boldsymbol{0}^n$. It is known that the probability that the infinite norm of $\boldsymbol{x} \leftarrow\!\!\$\, D_{\mathcal{L},\sigma}$ is less than $\sigma\sqrt{m}$ is greater than $1 - 2^{\Omega(m)}$ [7].

We use the following algorithms in our proposed scheme.

- $\mathsf{TrapGen}(1^n, 1^m, q)$ [1]: Given 1^n, 1^m and $q > 2$ with $m \geq \Omega(n \log q)$, it generates a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\boldsymbol{T}_{\boldsymbol{A}}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$ such that the distribution of $\boldsymbol{A}$ is $2^{-\Omega(n)}$-close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$ and $\left\|\tilde{\boldsymbol{T}_{\boldsymbol{A}}}\right\| \leq O(\sqrt{n \log q})$.

- $\mathsf{ExtBasis}(\overline{\boldsymbol{A}}, \boldsymbol{T}_{\boldsymbol{A}})$ [13]: Given $\overline{\boldsymbol{A}} \in \mathbb{Z}^{n \times \overline{m}}$ of the form $\overline{\boldsymbol{A}} = [\boldsymbol{A}|\boldsymbol{A}']$ and a basis $\boldsymbol{T}_{\boldsymbol{A}}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$, it returns a basis $\boldsymbol{T}_{\overline{\boldsymbol{A}}}$ of $\Lambda_q^{\perp}(\overline{\boldsymbol{A}})$ with $\left\|\boldsymbol{T}_{\overline{\boldsymbol{A}}}\right\| \leq \|\boldsymbol{T}_{\boldsymbol{A}}\|$.

- $\mathsf{SamplePre}(\boldsymbol{A}, \boldsymbol{T}_{\boldsymbol{A}}, \boldsymbol{u}, \sigma)$ [16]: Given $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, a basis $\boldsymbol{T}_{\boldsymbol{A}}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$, $\sigma \geq \left\|\tilde{\boldsymbol{T}_{\boldsymbol{A}}}\right\| \cdot \omega(\sqrt{\log m})$ and $\boldsymbol{u} \in \mathbb{Z}^n$, it returns $\boldsymbol{e} \leftarrow\!\!\$\, D_{\mathbb{Z}^m,\sigma}$ conditioned on $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \bmod q$.

  For convenience of expression, a matrix $\boldsymbol{U} = [\boldsymbol{u}_1 \mid \cdots \mid \boldsymbol{u}_m] \in \mathbb{Z}_q^{n \times m}$ can be given to $\mathsf{SamplePre}$ instead of a vector $\boldsymbol{u} \in \mathbb{Z}^n$. In this case, $\mathsf{SamplePre}$ generates a vector $\boldsymbol{e}_j \leftarrow\!\!\$\, \mathsf{SamplePre}(\boldsymbol{A}, \boldsymbol{T}_{\boldsymbol{A}}, \boldsymbol{u}_j, \sigma)$ for any $j \in [1, m]$, and then returns the matrix $\boldsymbol{E} = [\boldsymbol{e}_1 \mid \cdots \mid \boldsymbol{e}_m]$.

We employ the following cryptographic assumptions.

- Short Integer Solution ($\mathsf{SIS}_{n,m,q,\beta}$) Assumption: Let n, m, q and β be functions in $\lambda \in \mathbb{N}$. The $\mathsf{SIS}_{n,m,q,\beta}$ assumption states that any PPT adversary $\mathcal{A}$ can find a non-zero vector $\boldsymbol{x} \in \Lambda_q^{\perp}(\boldsymbol{A})$ with $\|\boldsymbol{x}\| \leq \beta$ with at most negligible probability in λ given a randomly chosen matrix $\boldsymbol{A} \leftarrow\!\!\$\, \mathbb{Z}_q^{n \times m}$.

- Decisional Learning with Errors ($\mathsf{LWE}_{n,q,\chi}$) Assumption: Let n and q be functions in λ, and let χ be a distribution over $\mathbb{Z}$. The $\mathsf{LWE}_{n,q,\chi}$ assumption states that any adversary $\mathcal{A}$ can distinguish whether given vectors $(\boldsymbol{A}, \boldsymbol{t})$ are generated by the following two distributions:
  - $\boldsymbol{A} \leftarrow\!\!\$\, \mathbb{Z}_q^{n \times m}; \boldsymbol{t} \leftarrow\!\!\$\, \mathbb{Z}_q^m$
  - $\boldsymbol{A} \leftarrow\!\!\$\, \mathbb{Z}_q^{n \times m}; \boldsymbol{s} \leftarrow\!\!\$\, \mathbb{Z}_q^n; \boldsymbol{e} \leftarrow\!\!\$\, \chi^m; \boldsymbol{t} \leftarrow \boldsymbol{A}^T\boldsymbol{s} + \boldsymbol{e}$.

We designate a B-bounded distribution as χ, namely χ samples a vector whose infinite norm is less than B, and a B'-bounded distribution as χ' for $B \geq \sqrt{n}\omega(\log n)$ and $B'(\lambda) = B(\lambda) \cdot \lambda^{\omega(1)}$. Concrete distributions were introduced in [29].

2.2 Group Signatures with Designated Traceability

We recap the notion of group signature with designated traceability (GSdT) [3]. Designating the openers is realized by specifying an access policy on attributes. Let $\mathbb{U}$ be an attribute universe. We write $Y(X) = 1$ to denote that an attribute X satisfies an access policy Y.

2.2.1 Syntax

Entities of GSdT are an issuer $\mathcal{I}$, an opening master $\mathcal{OM}$, openers $\mathcal{OP}_j$ and users $\mathcal{U}_i$. GSdT GSdT consists of the following algorithms and a protocol:

- $\mathsf{GKG}(1^\lambda, \mathbb{U})$: This is a PPT key generator. Given a security parameter 1^λ and an attribute universe $\mathbb{U}$, it returns a group public key gpk, an issuing key ik and an opening master key omk. Then, ik and omk are owned by $\mathcal{I}$ and $\mathcal{OM}$, respectively. Suppose that the registration table **reg** for the group is initialized.

- $\mathsf{OKG}(\mathsf{gpk}, \mathsf{omk}, j, X)$: This is a PPT opener key generator run by $\mathcal{OM}$. Given a group public key gpk, an opening master key omk, an index j of $\mathcal{OP}_j$ and $\mathcal{OP}_j$'s attribute X, it returns an opening key ok_j of $\mathcal{OP}_j$ with respect to the attribute X.

- $\mathsf{UKG}(1^\lambda)$: This is a PPT user key generator for $\mathcal{U}_i$ which is used to generate a key pair $(\mathsf{upk}, \mathsf{usk})$ that is required to join a group. Given a security parameter 1^λ, it returns a user public key upk and its user secret key usk. Suppose that upk is publicly certified e.g. in the PKI.

- Joining Protocol: When a user $\mathcal{U}_i$ of index i owning a key pair $(\mathsf{upk}_i, \mathsf{usk}_i)$ wants to join a group, the *interactive joining protocol* must be run between a user $\mathcal{U}_i$ and the issuer $\mathcal{I}$. Since the joining protocols of many dynamic group signatures [9, 3] have only 2 moves, we only consider 2 moves joining protocol like Fig. 1. The interfaces of the three algorithms used in the joining protocol are specified as $\mathsf{Join}_1(\mathsf{gpk}, i, \mathsf{upk}_i, \mathsf{usk}_i)$, $\mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk}_i, \mathbf{reg}, M_1)$ and $\mathsf{Join}_2(\mathsf{st}, M_2)$.

  Given a group public key gpk, $\mathcal{U}_i$'s index i and $\mathcal{U}_i$'s key pair $(\mathsf{upk}_i, \mathsf{usk}_i)$, Join_1 returns a state st and a first message M_1 sent to $\mathcal{I}$. Given a group public key gpk, an issuer key ik, $\mathcal{U}_i$'s index i, the user public key upk_i, the registration table **reg** and the first message M_1, Iss returns an updated registration table **reg** and a second message M_2 if the input tuple is appropriate. Given the state st and the second message M_2, Join_2 returns a group secret key gsk_i for $\mathcal{U}_i$. $\mathcal{U}_i$ stores his/her group secret key gsk_i and $\mathcal{I}$ stores the updated registration table **reg**.

- $\mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_i, Y, \mu)$: This is a PPT group signing algorithm run by a joined user $\mathcal{U}_i$ of index i with secret key gsk_i. Given a group public key gpk, the group secret key gsk_i, an access policy Y and a message μ, it returns a signature $\Sigma = (Y, \Sigma_0)$ of μ under gpk. The signer of Σ can be traced only by openers owning the attribute that satisfies Y.

- $\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, \Sigma_0))$: This is a DPT verifying algorithm. Given a group public key gpk, a message μ and a signature $\Sigma = (Y, \Sigma_0)$, it returns 1 if and only if Σ is a valid signature of μ under gpk.

- $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}_j, \mathbf{reg}, \mu, (Y, \Sigma_0))$: This is a PPT opening algorithm run by $\mathcal{OP}_j$ and opening key ok_j. Given a group public key gpk, the opening key ok_j with respect to the attribute X, a registration table **reg**, a message μ and a signature (Y, Σ_0), it returns an index i of the traced signer and a proof τ proving that the opening process is indeed honest.

- $\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, \mu, (Y, \Sigma_0), \tau)$: This is a DPT judging algorithm in the sense that it judges the honesty of an opening process. Given a group public key gpk, an index i of a traced user, user's public key upk_i, a message μ, a signature (Y, Σ_0) and a proof τ, it returns 1 if and only if τ is a valid with respect to the signature (Y, Σ_0) and the opening result (i, upk_i).

2.2.2 Security

The following four security notions were defined in [3]. Before that, we recap the oracles used in the definitions of these security notions.

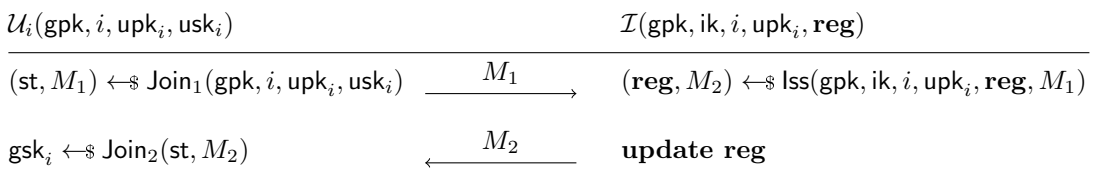

Figure 1: Joining Protocol

Oracles The stateful oracles used in the security notion are listed in the following way. The states are the honest openers set HO, the honest users set HU, the corrupted openers set CO, the corrupted users set CU, the user public key table $\mathbf{upk}$, the user secret key table $\mathbf{usk}$, the group secret key table $\mathbf{gsk}$, the opener key table $\mathbf{ok}$, the registration table $\mathbf{reg}$, the challenged message and signature set MS, the state list $\mathbf{st}_{\mathsf{Join}}$ for $(\mathsf{Join}_1, \mathsf{Join}_2)$ and the state list $\mathbf{st}_{\mathsf{Iss}}$ for Iss, respectively.

- Add-opener oracle $\mathsf{AddOO}(j, X)$: It registers a new opener $\mathcal{OP}_j$ of index j with the attribute X in a way that the honest openers set HO is updated to $HO \cup \{j\}$ and its opener key is generated as $\mathsf{ok}[j] \leftarrow \mathsf{OKG}(\mathsf{gpk}, \mathsf{omk}, j, X)$ if for any $(m, (Y, \Sigma_0)) \in MS$, $Y(X) \neq 1$.
- Add-user oracle $\mathsf{AddUO}(i)$: It registers a new user $\mathcal{U}_i$ of index $i \notin HU \cup CU$. Namely, it updates the honest users set HU as $HU \leftarrow HU \cup \{i\}$, generates user's key pair $(\mathbf{upk}[i], \mathbf{usk}[i]) \leftarrow\!\$\ \mathsf{UKG}(1^\lambda)$ and then runs the joining protocol by sequentially executing the followings:

$$\begin{aligned}(\mathsf{st}, M_1) &\leftarrow\!\$\ \mathsf{Join}_1(\mathsf{gpk}, i, \mathbf{upk}[i], \mathbf{usk}[i])\\ (\mathbf{reg}, M_2) &\leftarrow\!\$\ \mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, i, \mathbf{upk}[i], \mathbf{reg}, M_1)\\ \mathbf{gsk}[i] &\leftarrow\!\$\ \mathsf{Join}_2(\mathsf{st}, M_2)\end{aligned}$$

  Then, $\mathbf{st}_{\mathsf{Join}}[i] \leftarrow (\mathsf{gpk}, i, \mathbf{upk}[i], \mathbf{usk}[i])$ as the initialization of StoUO on i which will be explained soon later. Finally, AddUO returns $\mathbf{upk}[i]$.
- Send to user oracle $\mathsf{StoUO}(i, M_{in})$: If $i \notin HU$, then it initializes the user $\mathcal{U}_i$ of index i in a sense that it sets $HU \leftarrow HU \cup \{i\}$, $(\mathbf{upk}[i], \mathbf{usk}[i]) \leftarrow\!\$\ \mathsf{UKG}(1^\lambda)$, and $\mathbf{st}_{\mathsf{Join}}[i] \leftarrow (\mathsf{gpk}, i, \mathbf{upk}[i], \mathbf{usk}[i])$. When $M_{in} = \epsilon$, StoUO executes the followings:

$$\begin{aligned}(\mathsf{st}, M_{out}) &\leftarrow\!\$\ \mathsf{Join}_1(\mathsf{gpk}, i, \mathbf{upk}[i], \mathbf{usk}[i])\\ \mathbf{st}_{\mathsf{Join}}[i] &\leftarrow \mathsf{st}\end{aligned}$$

  Contrary, it executes the followings:

$$\begin{aligned}M_{out} &\leftarrow\!\$\ \mathsf{Join}_2(\mathbf{st}_{\mathsf{Join}}[i], M_{in})\\ \mathbf{gsk}[i] &\leftarrow M_{out}\\ \mathbf{st}_{\mathsf{Join}}[i] &\leftarrow (\mathsf{gpk}, i, \mathbf{upk}[i], \mathbf{usk}[i])\end{aligned}$$

  Either way, it returns M_{out}.
- Send to issuer oracle $\mathsf{StoIO}(i, M_{in})$: StoIO is executed only for a corrupted user, i.e. $i \in CU$. It executes $(\mathbf{reg}, M_{out}) \leftarrow\!\$\ \mathsf{Iss}(\mathsf{gpk}, \mathsf{ik}, i, \mathbf{upk}[i], \mathbf{reg}, M_{in})$, and then returns M_{out}.
- Corrupt opener oracle $\mathsf{CrptOO}(j)$: It returns the opener key $\mathbf{ok}[j]$ of the opener $\mathcal{OP}_j$ of index j if $j \in HO$ and for any $(m, (Y, \Sigma_0)) \in MS$, $Y(X) \neq 1$ for $(X, \mathsf{ok}_0) \leftarrow \mathbf{ok}[j]$. Then, CO is updated to $CO \cup \{j\}$.
- Corrupt user oracle $\mathsf{CrptUO}(i, \mathsf{upk})$: CrptUO corrupts a user $\mathcal{U}_i$ of index $i \notin HU \cup CU$ by setting $CU \leftarrow CU \cup \{i\}$, $\mathbf{upk}[i] \leftarrow \mathsf{upk}$ and $\mathbf{st}_{\mathsf{Iss}}[i] \leftarrow (\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk})$.
- User secret key oracle $\mathsf{USKO}(i)$: It returns the secret keys $(\mathbf{usk}[i], \mathbf{gsk}[i])$ of $\mathcal{U}_i$.

- Group signing oracle $\mathsf{GSignO}(i, Y, \mu)$: It returns a group signature (Y, Σ_0) of a given message μ under a given access policy Y by a group secret key $\mathbf{gsk}[i]$ of a given index $i \in HU$ of an honest user which has been already generated by using either AddUO or StoUO. More specifically, $(Y, \Sigma_0) \leftarrow\!\!\$\ \mathsf{GSig}(\mathsf{gpk}, \mathbf{gsk}[i], Y, \mu)$.
- Opening signature oracle $\mathsf{OpenO}(j, \mu, (Y, \Sigma_0))$: It returns the opening result from the opener $\mathcal{OP}_j$ of the given index j by $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}[j], \mu, (Y, \Sigma_0))$ only when the given pair $(\mu, (Y, \Sigma_0))$ does not belong to the set MS.
- Read registration table oracle $\mathsf{RRegO}(i)$: It returns $\mathbf{reg}[i]$.
- Write registration table oracle $\mathsf{WRegO}(i, \rho)$: WRegO sets $\mathbf{reg}[i] \leftarrow \rho$.
- Challenge for $b \in \{0, 1\}$ oracle $\mathsf{ChaO}_b(i_0, i_1, \mu, Y)$: For user's indices $i_0, i_1 \in HU$ satisfying that $\mathbf{gsk}[i_0] \neq \epsilon$ and $\mathbf{gsk}[i_1] \neq \epsilon$, and an access policy Y such that $Y(X) \neq 1$ for any $j \in HO \cup CO$ and $(X, \mathsf{ok}_0) \leftarrow \mathbf{ok}[j]$, ChaO_b returns $\Sigma \leftarrow\!\!\$\ \mathsf{GSig}(\mathsf{gpk}, \mathbf{gsk}[i_b], \mu)$ with updating $MS \leftarrow MS \cup \{(\mu, \Sigma)\}$.

Correctness GSdT is ν-*correct* [3] if for any PPT adversary $\mathcal{A}$,

$$\Pr\left[\mathsf{Exp}^{\mathrm{corr}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] \geq 1 - \nu(\lambda),$$

where $\mathsf{Exp}^{\mathrm{corr}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$ is depicted as follows:

$\mathsf{Exp}^{\mathrm{corr}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$

$(\mathsf{gpk}, \mathsf{ik}, \mathsf{omk}) \leftarrow\!\!\$\ \mathsf{GKG}(1^\lambda, \mathbb{U})$; $HO \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CO \leftarrow \emptyset$
$\mathbf{upk} \leftarrow \emptyset$; $\mathbf{usk} \leftarrow \emptyset$; $\mathbf{gsk} \leftarrow \emptyset$; $\mathbf{ok} \leftarrow \emptyset$; $\mathbf{reg} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Join}} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Iss}} \leftarrow \emptyset$
$(i, \mu, Y) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{AddOO},\mathsf{AddUO},\mathsf{RRegO}}(\mathsf{gpk})$
return 0 **if** $i \notin HU \vee \mathbf{gsk}[i] = \epsilon$
$\Sigma \leftarrow\!\!\$\ \mathsf{GSig}(\mathsf{gpk}, \mathbf{gsk}[i], Y, \mu)$
return 1 **if** $\mathsf{GVf}(\mathsf{gpk}, \mu, \Sigma) \neq 1$
$OS_Y \leftarrow \{j \in HO \mid Y(X) = 1 \textbf{ for } (X, \mathsf{ok}) \leftarrow \mathbf{ok}[j]\}$
for $j \in OS_Y$:
  $(i', \tau) \leftarrow\!\!\$\ \mathsf{Open}(\mathsf{gpk}, \mathbf{ok}[j], \mathbf{reg}, \mu, \Sigma)$
  return 1 **if** $i \neq i' \vee \mathsf{Judge}(\mathsf{gpk}, i, \mathbf{upk}[i], \mu, \Sigma, \mu) \neq 1$
return 0

Anonymity GSdT is $(T_{\mathrm{anom}}, \epsilon_{\mathrm{anom}})$-*anonymous* [3] if for any adversary $\mathcal{A}$ running in time T_{anom}, we have

$$\left|\Pr\left[\mathsf{Exp}^{\text{anom-0}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{anom-1}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}(\lambda) = 1\right]\right| \leq \epsilon_{\mathrm{anom}}(\lambda),$$

where $\mathsf{Exp}^{\text{anom-}b}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$ for $b \in \{0, 1\}$ is depicted as follows:

$\mathsf{Exp}^{\text{anom-}b}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$

$(\mathsf{gpk}, \mathsf{ik}, \mathsf{omk}) \leftarrow\!\!\$\ \mathsf{GKG}(1^\lambda, \mathbb{U})$; $HO \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CO \leftarrow \emptyset$; $CU \leftarrow \emptyset$
$\mathbf{upk} \leftarrow \emptyset$; $\mathbf{usk} \leftarrow \emptyset$; $\mathbf{gsk} \leftarrow \emptyset$; $\mathbf{ok} \leftarrow \emptyset$; $\mathbf{reg} \leftarrow \emptyset$; $MS \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Join}} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Iss}} \leftarrow \emptyset$
$d \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{ChaO}_b,\mathsf{AddOO},\mathsf{OpenO},\mathsf{StoUO},\mathsf{WRegO},\mathsf{USKO},\mathsf{CrptOO},\mathsf{CrptUO}}(\mathsf{gpk}, \mathsf{ik})$
return d

Traceability GSdT is $(T_{\mathrm{trac}}, \epsilon_{\mathrm{trac}})$-*traceable* [3] if for any adversary $\mathcal{A}$ running in time T_{trac}, we have

$$\Pr\left[\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] \leq \epsilon_{\mathrm{trac}}(\lambda),$$

where $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$ is depicted as follows:

$\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$

$(\mathsf{gpk}, \mathsf{ik}, \mathsf{omk}) \leftarrow\!\!\$\ \mathsf{GKG}(1^\lambda, \mathbb{U})$; $HO \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CO \leftarrow \emptyset$
$\mathbf{upk} \leftarrow \emptyset$; $\mathbf{usk} \leftarrow \emptyset$; $\mathbf{gsk} \leftarrow \emptyset$; $\mathbf{ok} \leftarrow \emptyset$; $\mathbf{reg} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Join}} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Iss}} \leftarrow \emptyset$
$(\mu, (Y, \Sigma_0)) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{StoIO},\mathsf{AddUO},\mathsf{RRegO},\mathsf{USKO},\mathsf{CrptUO}}(\mathsf{gpk}, \mathsf{omk})$
return 0 **if** $\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, \Sigma_0)) \neq 1$
find X **s.t.** $Y(X) = 1$; $\mathsf{ok} \leftarrow\!\!\$\ \mathsf{OKG}(\mathsf{gpk}, \mathsf{omk}, 0, X)$
$(i, \tau) \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathbf{reg}, \mu, (Y, \Sigma_0))$
return 1 **if** $i = 0 \vee \mathsf{Judge}(\mathsf{gpk}, i, \mathbf{upk}[i], \mu, (Y, \Sigma_0), \tau) \neq 1$
else return 0

Non-frameability GSdT is $(T_{\mathrm{nf}}, \epsilon_{\mathrm{nf}})$-*non-frameable* [3] if for any adversary $\mathcal{A}$ running in time T_{nf}, we have

$$\Pr\left[\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] \leq \epsilon_{\mathrm{nf}}(\lambda),$$

where $\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$ is depicted as follows:

$\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT},\mathcal{A},\mathbb{U}}$

$(\mathsf{gpk}, \mathsf{ik}, \mathsf{omk}) \leftarrow\!\!\$\ \mathsf{GKG}(1^\lambda, \mathbb{U})$; $HO \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CO \leftarrow \emptyset$
$\mathbf{upk} \leftarrow \emptyset$; $\mathbf{usk} \leftarrow \emptyset$; $\mathbf{gsk} \leftarrow \emptyset$; $\mathbf{ok} \leftarrow \emptyset$; $\mathbf{reg} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Join}} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Iss}} \leftarrow \emptyset$
$(\mu, (Y, \Sigma_0), i, \tau) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{StoUO},\mathsf{WRegO},\mathsf{GSignO},\mathsf{USKO},\mathsf{CrptUO}}(\mathsf{gpk}, \mathsf{ik}, \mathsf{omk})$
return 1 **if** $i \in HU \wedge \mathbf{gsk}[i] \neq \epsilon \wedge \mathsf{Judge}(\mathsf{gpk}, i, \mathbf{upk}[i], \mu, (Y, \Sigma_0), \tau) = 1$
$\wedge\ i$ is not queried to $\mathsf{USKO} \wedge (i, \mu)$ is not queried to GSignO
else return 0

3 Building Blocks

Before introducing our proposed GSdT, we prepare several sub-algorithms of the proposed lattice-based GSdT. The sub-algorithms contain the lattice-based signature scheme DS by [11], the Stern-like non-interactive zero-knowledge argument (NIZKAoK) NIZK by [22], and the lattice-based ciphertext-policy attribute-based encryption (CP-ABE) ABE by [29].

3.1 Digital Signature Part DS

3.1.1 Definition of Digital Signature

A digital signature DS consists of the following three algorithms.

- $\mathsf{SKGen}(1^\lambda)$: This is a PPT key generator. Given a security parameter 1^λ, it returns a public key pk and a corresponding secret key sk.
- $\mathsf{Sign}(\mathsf{sk}, \mu)$: This is a PPT signing algorithm. Given a secret key sk and a message μ, it returns a signature Σ.

$\underline{\mathsf{SKGen}(1^\lambda)}$

$(\boldsymbol{A}, \boldsymbol{T_A}) \leftarrow\!\$\ \mathsf{TrapGen}(1^n, 1^m, q)$
for $t \in [1, \ell] : \boldsymbol{A}_t \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}$
$\boldsymbol{u} \leftarrow\!\$\ \mathbb{Z}_q^n$
$\boldsymbol{D} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}; \quad \boldsymbol{D}_0, \boldsymbol{D}_1 \leftarrow\!\$\ \mathbb{Z}_q^{2n\times 2m}$
$\mathsf{pk} \leftarrow (\boldsymbol{A}, \{\boldsymbol{A}_t\}_{t\in[1,\ell]}, \boldsymbol{D}, \boldsymbol{D}_0, \boldsymbol{D}_1, \boldsymbol{u})$
$\mathsf{sk} \leftarrow \boldsymbol{T_A}$
return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Sign}(\mathsf{sk}, \mu)}$

$\tau = (\tau[1], \tau[2], \ldots, \tau[\ell]) \leftarrow\!\$\ \{0,1\}^\ell$
$\boldsymbol{A}_\tau \leftarrow [\ \boldsymbol{A} \ \big|\ \boldsymbol{A}_0 + \sum_{t=1}^{\ell} \tau[t]\boldsymbol{A}_t\ ]$
$\boldsymbol{T}_\tau \leftarrow\!\$\ \mathsf{ExtBasis}(\boldsymbol{A}_\tau, \boldsymbol{T_A})$
$\boldsymbol{s} \leftarrow\!\$\ D_{\mathbb{Z}^{2m},\sigma'}; \quad \boldsymbol{c} \leftarrow \boldsymbol{D}_0\boldsymbol{s} + \boldsymbol{D}_1\mu$
$\boldsymbol{\gamma} \leftarrow \mathsf{bin}(\boldsymbol{c}); \quad \boldsymbol{u}_\mu \leftarrow \boldsymbol{u} + \boldsymbol{D}\cdot\boldsymbol{\gamma}$
$\boldsymbol{d} \leftarrow\!\$\ \mathsf{SamplePre}(\boldsymbol{A}_\tau, \boldsymbol{T}_\tau, \boldsymbol{u}_\mu, \sigma)$
return $\Sigma \leftarrow (\tau, \boldsymbol{d}, \boldsymbol{s})$

$\underline{\mathsf{Vf}(\mathsf{pk}, \mu, \Sigma)}$

$\boldsymbol{u}_\mu \leftarrow \boldsymbol{u} + \boldsymbol{D}\cdot\mathsf{bin}(\boldsymbol{D}_0\boldsymbol{s} + \boldsymbol{D}_1\mu)$
return 1 **if** $\|\boldsymbol{d}\| < \sigma\sqrt{2m} \wedge \|\boldsymbol{s}\| < \sigma'\sqrt{2m} \wedge \boldsymbol{A}_\tau\boldsymbol{d} = \boldsymbol{u}_\mu \bmod q$

Figure 2: Digital Signature Part DS (The message space is $\{0,1\}^{2m}$)

- $\mathsf{Vf}(\mathsf{pk}, \mu, \Sigma)$: This is a DPT verification algorithm. Given a public key pk, a message μ and a signature Σ, it returns 1 if and only if Σ is valid under (pk, μ).

$\mathsf{DS} = (\mathsf{SKGen}, \mathsf{Sign}, \mathsf{Vf})$ has the following properties.

Correctness: For any $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{SKGen}(1^\lambda)$, any message μ, and any signature $\Sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, \mu)$, it always holds that $\mathsf{Vf}(\mathsf{pk}, \mu, \Sigma) = 1$.

EUF-CMA**:** We say that DS *is* $(T_{\mathsf{DS}}, \epsilon_{\mathsf{DS}})$-EUF-CMA if for any adversary $\mathcal{A}$ running in time T_{DS}, we have

$$\Pr\left[\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{A}}(\lambda) = 1\right] \le \epsilon_{\mathsf{DS}}(\lambda),$$

where $\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{A}}$ is formalized as follows:

$\underline{\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{A}}}$

$M \leftarrow \emptyset; \quad (\mathsf{sk}, \mathsf{pk}) \leftarrow\!\$\ \mathsf{SKGen}(1^\lambda)$
$(\mu^*, \Sigma^*) \leftarrow\!\$\ \mathcal{A}^{\mathsf{Osig}}(\mathsf{pk})$
return 1 **if** $\mu^* \notin M \wedge \mathsf{Vf}(\mathsf{pk}, \mu^*, \Sigma^*) = 1$

$\underline{\mathsf{Osig}(\mu)}$

$M \leftarrow M \cup \{\mu\}$
return $\Sigma \leftarrow\!\$\ \mathsf{Sign}(\mathsf{sk}, \mu)$

3.1.2 BHJKS DS

Our underlying signature $\mathsf{DS} = (\mathsf{SKGen}, \mathsf{Sign}, \mathsf{Vf})$ is given in Fig. 2 with the parameters in Tab. 1.

Lemma 1 ([11])**.** DS *is correct, and it is* (T, ϵ)-EUF-CMA *under the* $\mathsf{SIS}_{n,m,q,\beta}$ *assumption, where* $T = \mathsf{poly}(\lambda)$ *and* $\epsilon = \mathsf{negl}(\lambda)$.

3.2 NIZKAoK Part NIZK

3.2.1 Definition of NIZK

Let $R_{\mathsf{NIZK}} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation. Suppose that the membership of R_{NIZK} can be verified in polynomial-time on the length of a pair $(\mathsf{it}, \mathsf{wt})$. In this paper, we only consider a non-interactive zero-knowledge proof of knowledge (NIZK) in the *random oracle model*. A labeled NIZK $\mathsf{NIZK} = (P, V)$ for R_{NIZK} consists of the following two algorithms:

- $P(\mathsf{it}, \mathsf{wt}, \mathrm{lbl})$: It is a PPT prover algorithm. Given an instance it, its witness wt such that $(\mathsf{it}, \mathsf{wt}) \in R_{\mathsf{NIZK}}$ and a label string $\mathrm{lbl} \in \{0,1\}^*$, it returns a proof π.
- $V(\mathsf{it}, \pi, \mathrm{lbl})$: It is a DPT verifier algorithm. Given an instance it, a proof π and a label string lbl, it returns 1 if and only if π is valid.

$\mathsf{NIZK} = (P, V)$ has the following properties [15], where $L_{R_{\mathsf{NIZK}}} = \{\mathsf{it} \mid \exists \mathsf{wt}$ **s.t.** $(\mathsf{it}, \mathsf{wt}) \in R_{\mathsf{NIZK}}\}$:

Perfect Completeness For any $(\mathsf{it}, \mathsf{wt}) \in R_{\mathsf{NIZK}}$, any label string $\mathrm{lbl} \in \{0,1\}^*$, and any $\pi \leftarrow\!\!\$\ P(\mathsf{it}, \mathsf{wt}, \mathrm{lbl})$ we have $V(\mathsf{it}, \pi, \mathrm{lbl}) = 1$.

(T_s, ϵ_s)-Soundness For any $\mathsf{it} \notin L_{R_{\mathsf{NIZK}}}$, any adversary $\tilde{P}$ running in time T_s, any label string lbl, and any cheating proof $\pi \leftarrow\!\!\$\ \tilde{P}(\mathsf{it}, \mathrm{lbl})$, we have $V(\mathsf{it}, \pi, \mathrm{lbl}) = 1$ with probability at most ϵ_s.

$(T_{\mathsf{zk}}, \epsilon_{\mathsf{zk}})$-Zero-Knowledge in Random Oracle Model There exist a PPT simulation algorithm $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_P)$ such that for any PPT adversary $\mathcal{A}$ running in time T_{zk}, it holds that

$$\left|\Pr\left[\mathcal{A}^{\mathsf{H}, P^{\mathsf{H}}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{Sim}_H, \mathsf{Sim}_P}(1^\lambda) = 1\right]\right| \leq \epsilon_{\mathsf{zk}}(\lambda),$$

where H denotes the random oracle, and $P^{\mathsf{H}}(\mathsf{it}, \mathrm{lbl})$ returns $\pi \leftarrow\!\!\$\ P(\mathsf{it}, \mathsf{wt}, \mathrm{lbl})$ for the witness wt corresponding to the given instance it.

$(T_{\mathsf{ss}}, \epsilon_{\mathsf{ss}})$-Simulation-Soundness with respect to Simulator $(\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_P)$ For any adversary $\mathcal{A}$ running in time T_{ss}, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{ss}}_{\mathsf{NIZK}, \mathcal{A}}(\lambda) = 1\right] \leq \epsilon_{\mathsf{ss}}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{ss}}_{\mathsf{NIZK}, \mathcal{A}}$ is depicted as follows:

$\mathsf{Exp}^{\mathsf{ss}}_{\mathsf{NIZK}, \mathcal{A}}$

$(\mathsf{it}, \mathrm{lbl}, \pi) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Sim}_{\mathsf{H}}, \mathsf{OP}}(1^\lambda)$

return 1 **if** $\mathsf{it} \notin L R_{\mathsf{NIZK}} \wedge (\mathsf{it}, \mathrm{lbl}, \pi) \notin T \wedge V^{\mathsf{Sim}_{\mathsf{H}}}(\mathsf{it}, \mathrm{lbl}, \pi) = 1$

$\mathsf{OP}(\mathsf{it}, \mathrm{lbl})$

return $\pi \leftarrow\!\!\$\ \mathsf{Sim}_P(\mathsf{it}, \mathrm{lbl})$

$T \leftarrow T \cup \{(\mathsf{it}, \mathrm{lbl}, \pi)\}$

$(T_{\mathsf{ext}}, \nu_{\mathsf{ext}})$-Weak-Simulation-Extractability with respect to Simulator $(\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_P)$ For any adversary $\mathcal{A}$ running in time T_{ext}, there exist an extraction algorithm $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$, a constant $d > 0$ and a polynomial p such that

$$ext \geq \frac{1}{p}(acc - \nu_{\mathsf{ext}})^d,$$

where

$$acc = \Pr_{r \leftarrow\!\!\$ \{0,1\}^{\ell(\lambda)}}\left[\mathsf{it} \in L_{R_{\mathsf{NIZK}}} \wedge V^{\mathsf{Sim}_{\mathsf{H}}}(\mathsf{it}, \mathrm{lbl}, \pi) = 1 \mid (\mathsf{it}, \mathrm{lbl}, \pi) \leftarrow \mathcal{A}^{\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_P}(1^\lambda; r)\right]$$

$$ext = \Pr_{r \leftarrow\!\!\$ \{0,1\}^{\ell(\lambda)}}\left[(\mathsf{it}, \mathsf{wt}) \in R_{\mathsf{NIZK}} \wedge V^{\mathsf{Sim}_{\mathsf{H}}}(\mathsf{it}, \mathrm{lbl}, \pi) = 1 \mid \begin{array}{l}(st, \mathsf{it}, \mathrm{lbl}, \pi) \leftarrow \mathsf{Ext}_1^{\mathcal{A}^{\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_P}}(1^\lambda; r); \\ \mathsf{wt} \leftarrow \mathsf{Ext}_2(st, \mathsf{it}, \mathrm{lbl}, \pi, r)\end{array}\right].$$

ϵ_{wi}-statistical-Witness-Indistinguishability [17] For instance $\mathsf{it} \in L_{\mathsf{NIZK}}$ and any distinct witness $\mathsf{wt}_1, \mathsf{wt}_2$ of it any unbounded adversary $\mathcal{A}$, we have

$$\left|\mathcal{A}^{P(\mathsf{it}, \mathsf{wt}_1)}(1^\lambda) - \mathcal{A}^{P(\mathsf{it}, \mathsf{wt}_2)}(1^\lambda)\right| \leq \epsilon_{\mathsf{wi}}(\lambda).$$

$P((\boldsymbol{P},\boldsymbol{v}),\boldsymbol{x},\mathrm{lbl})$

for $j \in [1,\kappa]$:

$\boldsymbol{r}_j \leftarrow\!\!\$\ \mathbb{Z}_q^L; \quad \boldsymbol{y}_j \leftarrow \boldsymbol{x}+\boldsymbol{r}_j; \quad \pi_j \leftarrow\!\!\$\ \mathcal{S}; \quad \rho_{j,1},\rho_{j,2},\rho_{j,3} \leftarrow\!\!\$\ \{0,1\}^\eta$

$$\mathsf{cmt}_j = \begin{bmatrix} C_{j,1} \\ C_{j,2} \\ C_{j,3} \end{bmatrix} \leftarrow \begin{bmatrix} \mathsf{COM}(\pi_j, \boldsymbol{P}\boldsymbol{r}_j; \rho_{j,1}) \\ \mathsf{COM}(T_{\pi_j}(\boldsymbol{r}_j); \rho_{j,2}) \\ \mathsf{COM}(T_{\pi_j}(\boldsymbol{y}_j); \rho_{j,3}) \end{bmatrix}$$

$\{\mathsf{cha}_j\}_{j\in[1,\kappa]} \leftarrow \mathsf{H}(\{\mathsf{cmt}_j\}_{j\in[1,\kappa]}, (\boldsymbol{P},\boldsymbol{v}), \mathrm{lbl})$

for $j \in [1,\kappa]$:

$$\mathsf{res}_j = (R_j, R'_j, \rho_j, \rho'_j) \leftarrow \begin{cases} (T_{\pi_j}(\boldsymbol{x}), T_{\pi_j}(\boldsymbol{r}_j), \rho_{j,2}, \rho_{j,3}) & \mathsf{cha}_j = 1 \\ (\pi_j, \boldsymbol{y}_j, \rho_{j,1}, \rho_{j,3}) & \mathsf{cha}_j = 2 \\ (\pi_j, \boldsymbol{r}_j, \rho_{j,1}, \rho_{j,2}) & \mathsf{cha}_j = 3 \end{cases}$$

return $\pi = \{(\mathsf{cmt}_j, \mathsf{res}_j)\}_{j\in[1,\kappa]}$

$V((\boldsymbol{P},\boldsymbol{v}), \{(\mathsf{cmt}_j, \mathsf{res}_j)\}_{j\in[1,\kappa]}, \mathrm{lbl})$

$\{\mathsf{cha}_j\}_{j\in[1,\kappa]} \leftarrow \mathsf{H}(\{\mathsf{cmt}_j\}_{j\in[1,\kappa]}, (\boldsymbol{P},\boldsymbol{v}), \mathrm{lbl})$

$\{((C_{j,1},C_{j,2},C_{j,3}),(r_j,r'_j,\rho_j,\rho'_j))\} \leftarrow \{(\mathsf{cmt}_j, \mathsf{res}_j)\}_{j\in[1,\kappa]}$

for $j \in [1,\kappa]$:

$$\textbf{return } 0 \textbf{ if } \begin{cases} r_j \notin \mathsf{W} \vee C_{j,2} \neq \mathsf{COM}(r'_j;\rho_j) \vee C_{j,3} \neq \mathsf{COM}(r_j + r'_j; \rho'_j) & \mathsf{cha}_j = 1 \\ C_{j,1} \neq \mathsf{COM}(T_{r_j}, \boldsymbol{P}r'_j - \boldsymbol{v}; \rho_j) \wedge C_{j,3} \neq \mathsf{COM}(T_{r_j}(r'_j); \rho'_j) & \mathsf{cha}_j = 2 \\ C_{j,1} \neq \mathsf{COM}(T_{r_j}, \boldsymbol{P}r'_j; \rho_j) \wedge C_{j,2} \neq \mathsf{COM}(T_{r_j}(r'_j); \rho'_j) & \mathsf{cha}_j = 3 \end{cases}$$

return 1

Figure 3: (Labeled) NIZK part NIZK

3.2.2 Abstract Stern NIZK

Binary Relation Let T_π be a permutation parameterized by $\pi \in \mathcal{S}$ in a sense that for any $\pi \in \mathcal{S}$, $T_\pi : \{-1,0,1\}^L \to \{-1,0,1\}^L$ is a permutation. The relation for NIZK is

$$R_{\mathsf{NIZK}} = \big\{((\boldsymbol{P},\boldsymbol{v}),\boldsymbol{x}) \in \mathbb{Z}_q^{D\times L} \times \mathbb{Z}_q^D \times \mathsf{W} : \boldsymbol{P}\boldsymbol{x} = \boldsymbol{v} \bmod q\big\},$$

where, W is a subset of $\{-1,0,1\}^L$ satisfying that

- for any $\pi \in \mathcal{S}$, $\boldsymbol{x} \in \mathsf{W}$ if and only if $T_\pi(\boldsymbol{x}) \in \mathsf{W}$, and
- if $\boldsymbol{x} \in \mathsf{W} \wedge \pi \leftarrow\!\!\$\ \mathcal{S}$, then $T_\pi(\boldsymbol{x})$ is uniformly distributed over W.

Protocol of NIZK A NIZK $\mathsf{NIZK} = (P,V)$ employed in our proposed GSdT is given in Fig. 3, where COM is the SIS-based commitment scheme [18], $\mathsf{H} : \{0,1\}^* \to \{1,2,3\}$ is a hash function, and κ denotes the iteration time.

Lemma 2 ([11, 27])**.** $\mathsf{NIZK} = (P,V)$ *is perfectly complete,* (T_s,ϵ_s)*-sound,* $(T_{\mathsf{zk}},\epsilon_{\mathsf{zk}})$*-zero-knowledge in the random oracle model for the simulator* $(\mathsf{Sim}_{\mathsf{H}},\mathsf{Sim}_P)$*,* $(T_{\mathsf{ss}},\epsilon_{\mathsf{ss}})$*-simulation-sound with respect to* $(\mathsf{Sim}_{\mathsf{H}},\mathsf{Sim}_P)$*, and* $(T_{\mathsf{ext}},\nu_{\mathsf{ext}})$*-weakly-simulation-extractable with respect to* $(\mathsf{Sim}_{\mathsf{H}},\mathsf{Sim}_P)$*,* ϵ_{wi}*-statistically-witness-indistinguishable, where* T_s*,* T_{zk}*,* T_{ss} *and* T_{ext} *are polynomials, and* ϵ_s*,* ϵ_{zk}*,* ϵ_{ss}*,* ν_{ext} *and* ϵ_{wi} *are negligible in* λ*.*

Representation of Instances and Witnesses A witness $\boldsymbol{x}$ proven by NIZK should be in W. On the other hand, we employ NIZK to prove the knowledge of elements in $[\pm B]$ for some constant B and bit strings. [22] proposed conversions for the following types:

(Type 1) $\boldsymbol{x} \in [\pm B]^m$ into W

(Type 2) $\boldsymbol{x} \in \{0,1\}^m$ into W

(Type 3) $\begin{bmatrix} \boldsymbol{d} \cdot \boldsymbol{b}[1] \\ \hline \vdots \\ \hline \boldsymbol{d} \cdot \boldsymbol{b}[n] \end{bmatrix} \in [\pm\beta]^{mn}$ for $(\boldsymbol{d}, \boldsymbol{b}) \in [\pm\beta]^m \times \{0,1\}^n$ into W

We recap the conversion of such elements into the ones in W. For any $m \in \mathbb{N}$, let $\mathcal{S}_m$ be the symmetric group of order m. For any $\boldsymbol{d} \in [\pm\beta]^m$ and any $\boldsymbol{b} \in \{0,1\}^n$, we defined $\boldsymbol{d}^{\boldsymbol{b}}$ as

$$\begin{bmatrix} \boldsymbol{d} \cdot \boldsymbol{b}[1] \\ \hline \vdots \\ \hline \boldsymbol{d} \cdot \boldsymbol{b}[n] \end{bmatrix} \in [\pm\beta]^{mn}.$$

(Type 1) Conversion of $\boldsymbol{x} \in [\pm B]^m$ into W Let $\delta_B = \lfloor \log_2 B \rfloor + 1$, and let $B^{(3)}_{m\delta_B}$ be the set of all vectors $\{0,1\}^{3m\delta_B}$ whose $m\delta_B$ elements are j for each $j \in \{-1, 0, 1\}$. There exists an algorithm $\mathsf{DecExt}_{m,B}$ that converts $\boldsymbol{x} = (x_1, \ldots, x_m)^T \in \{0,1\}^m$ into $\hat{\boldsymbol{x}} \in B^{(3)}_{m\delta_B}$ [22].

(Type 2) Conversion of $\boldsymbol{x} \in \{0,1\}^m$ into W Let $B^{(2)}_m$ be the set of all vectors in $\{0,1\}^{2m}$ such that m elements in each vector are 1, and the others are 0. Then, an algorithm Ext converts $\boldsymbol{x} \in \{0,1\}^m$ into $\begin{bmatrix} \boldsymbol{x} \\ \hline \overline{\boldsymbol{x}} \end{bmatrix} \in B^{(2)}_m$. We can see that $\hat{\boldsymbol{x}} \in B^{(2)}_m$ if and only if $\rho(\hat{\boldsymbol{x}}) \in B^{(2)}_m$ for any permutation $\rho \in \mathcal{S}_{2m}$. By defining the permutation $T_\rho(\hat{\boldsymbol{x}}) = \rho(\hat{\boldsymbol{x}})$ for each $\rho \in \mathcal{S}_{2m}$, $B^{(2)}_m$ with the set $\mathcal{S}_{2m}$ satisfies the conditions on W.

In a similar manner to the above case, we can observe that $B^{(3)}_{m\delta_B}$ with the set $\phi \in \mathcal{S}_{3\delta_B}$ satisfies the conditions on W.

(Type 3) Conversion of $\boldsymbol{d}^{\boldsymbol{b}}$ for $(\boldsymbol{d}, \boldsymbol{b}) \in [\pm\beta]^m \times \{0,1\}^n$ An algorithm $\mathsf{IDecExt}_{n,m,\beta}$ converts $(\boldsymbol{d}, \boldsymbol{b})$ into $\hat{\boldsymbol{d}}^{\hat{\boldsymbol{b}}} \in B^{(4)}_{n,m,\beta}$ for $\hat{\boldsymbol{d}} \leftarrow \mathsf{DecExt}_{m,\delta_B}(\boldsymbol{d})$ and $\hat{\boldsymbol{b}} \leftarrow \mathsf{Ext}_m(\boldsymbol{b})$, where

$$B^{(4)}_{n,m,\beta} = \{\hat{\boldsymbol{d}}^{\hat{\boldsymbol{b}}} \mid \boldsymbol{d} \in [\pm\beta]^m, \boldsymbol{b} \in \{0,1\}^n, \hat{\boldsymbol{d}} \leftarrow \mathsf{DecExt}_{m,\delta_B}(\boldsymbol{d}), \hat{\boldsymbol{b}} \leftarrow \mathsf{Ext}_m(\boldsymbol{b})\}.$$

Then, for $(\psi, \rho) \in \mathsf{S}_{3m\delta_B} \times \mathsf{S}_{2n}$, we define the parameterized permutation $T_{(\psi,\rho)}$ by

$$T_{(\psi,\rho)}\left(\begin{bmatrix} \hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{b}}[1] \\ \hline \vdots \\ \hline \hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{b}}[2n] \end{bmatrix}\right) = \begin{bmatrix} \psi(\hat{\boldsymbol{d}}) \cdot \hat{\boldsymbol{b}}[\rho(1)] \\ \hline \vdots \\ \hline \psi(\hat{\boldsymbol{d}}) \cdot \hat{\boldsymbol{b}}[\rho(2n)] \end{bmatrix}.$$

$B^{(4)}_{n,m,\beta}$ with the parameterized permutation $T_{(\psi,\rho)}$ satisfies the conditions W.

3.3 CP-ABE part ABE

Suppose that for $t \leq \xi$, any attribute X is represented by a ξ-bit string, and any access policy Y is a conjunctive normal form whose clauses have t bits of input (t-CNF), respectively.

3.3.1 Definition of CP-ABE

A CP-ABE ABE consists of the following polynomial-time algorithms:

- $\mathsf{APgen}(1^\lambda)$: This is a PPT parameter generator. Given a security parameter 1^λ, it returns a public key pk and a master secret key msk.

- $\mathsf{AKGen}(msk, X)$: This is a key generator. Given a master secret key msk and an attribute X, it returns a secret key sk_X over the attribute X.
- $\mathsf{Enc}(\mathsf{pk}, Y, \mu)$: This is a PPT encryption algorithm. Given a public key pk, an access policy Y over decrypters' attribute and a message μ, it returns a ciphertext cp of the message μ.
- $\mathsf{Dec}(\mathsf{sk}_X, \mathsf{cp})$: This is a DPT decryption algorithm. Given a secret key sk_X and a ciphertext cp, it returns either a decryption result μ' or the failure symbol $\perp$.

We will require that cp visibly contains Y, which is typical in the class of only-payload-hiding CP-ABE schemes [21].

$\mathsf{ABE} = (\mathsf{APgen}, \mathsf{AKGen}, \mathsf{Enc}, \mathsf{Dec})$ has the following properties:

Correctness: For any security parameter λ, any attribute X, any access policy Y, any message μ, any pair $(\mathsf{pk}, msk) \leftarrow\!\!\$\ \mathsf{APgen}(\lambda)$, any secret key $\mathsf{sk}_X \leftarrow\!\!\$\ \mathsf{AKGen}(msk, X)$, any ciphertext $\mathsf{cp} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, Y, \mu)$, and any decryption result $\mu' \leftarrow\!\!\$\ \mathsf{Dec}(\mathsf{sk}_X, \mathsf{cp})$, it holds that $\mu' = \mu$ if $Y(X) = 1$.

IND-CPA**:** We say that ABE *is* $(T_{\mathsf{ABE}}, \epsilon_{\mathsf{ABE}})$-IND-CPA if for any adversary $\mathcal{A}$ running in time at most T_{ABE}, it holds that

$$\left|\Pr\left[\mathsf{Exp}^{\text{IND-CPA-0}}_{\mathsf{ABE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{IND-CPA-1}}_{\mathsf{ABE},\mathcal{A}}(\lambda) = 1\right]\right| \leq \epsilon_{\mathsf{ABE}}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{ABE}\text{-}d}_{\mathsf{ABE},\mathcal{A}}$ is formalized as follows:

$\mathsf{Exp}^{\mathsf{ABE}\text{-}d}_{\mathsf{ABE},\mathcal{A}}(\lambda)$	$\mathsf{Ocorr}(X)$	$\mathsf{Och}\text{-}d(Y^*, \mu_0, \mu_1)$
$K \leftarrow \emptyset$	**return** $\perp$ **if** $\mathcal{R}(X, Y^*) = 1$	**return** $\perp$ **if** $\lvert\mu_0\rvert \neq \lvert\mu_1\rvert$
$(msk, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{APgen}(1^\lambda)$	$K \leftarrow K \cup \{X\}$	**return** $\perp$ **if** $\exists X \in K$ **s.t.** $Y(X) = 1$
return $\mathcal{A}^{\mathsf{Ocorr},\mathsf{Och}\text{-}d}(\mathsf{pk})$	**return** $\mathsf{AKGen}(msk, X)$	**return** $\mathsf{Enc}(\mathsf{pk}, Y^*, \mu_d)$

3.3.2 Conforming Constrained Pseudo-Random Function

To realize a lattice-based ABE for t-CNF, a special primitive called *conforming constrained pseudo-random function* (ccPRF) is employed [29]. A ccPRF is intuitively an extended notion of the constrained PRF which has not only the same properties as the ordinary PRFs, but also another property that a key ek_Y constrained by the boolean function Y can be generated by using an ordinary evaluation key mek of the PRF, and then ek_Y can be used to evaluate a value of the PRF on input X only when $Y(X) = 1$. A ccPRF ccPRF formally consists of the following polynomial-time algorithms with the parameters as in Tab. 1.

- $\mathsf{Pgen}(1^\lambda)$ returns a public parameter pp and a master evaluation key $\mathsf{mek} \in \{0,1\}^\lambda$.
- $\mathsf{Eval}(\mathsf{pp}, \mathsf{mek}, X)$ returns an evaluated value $\rho_X \in \{0,1\}^\lambda$ for any string $X \in \{0,1\}^\xi$.
- $\mathsf{Constrain}(\mathsf{pp}, \mathsf{mek}, Y)$ returns a constraining key $\mathsf{ek}_Y \in \{0,1\}^\iota$ under a given boolean function Y.
- $\mathsf{CEval}(\mathsf{pp}, \mathsf{ek}_Y, X)$ returns either an evaluated value $\rho'_X \in \{0,1\}^\lambda$ or the failure symbol $\perp$.
- $\mathsf{KSim}(\mathsf{pp}, Y)$ returns a faked key $\mathsf{ek}_Y \in \{0,1\}^\iota$ under Y.

$\mathsf{ccPRF} = (\mathsf{Pgen}, \mathsf{Eval}, \mathsf{Constrain}, \mathsf{CEval})$ has the following properties:

Correctness For any string $X \in \{0,1\}^\xi$ and any boolean function $Y : \{0,1\}^\xi \to \{0,1\}$, any pair $(\mathsf{pp}, \mathsf{mek}) \leftarrow\!\!\$\ \mathsf{Pgen}(1^\lambda)$ and any regular constraining key $\mathsf{ek}_Y \leftarrow \mathsf{Constrain}(\mathsf{pp}, \mathsf{mek}, Y)$, we have

$$\mathsf{CEval}(\mathsf{pp}, \mathsf{ek}_Y, X) = \begin{cases} \mathsf{Eval}(\mathsf{pp}, \mathsf{mek}, X) & Y(X) = 1, \\ \bot & Y(X) = 0. \end{cases}$$

Gradual Evaluation Suppose that for any $(\mathsf{pp}, \mathsf{mek}) \leftarrow\!\!\$\ \mathsf{Pgen}(1^\lambda)$, any string $X \in \{0,1\}^\xi$ and any boolean function $Y : \{0,1\}^\xi \to \{0,1\}$, there exist circuits $U_{\eta\to X}$, $U_{\eta\to Y}$ and $U_{Y\to X}$ such that

$$\begin{aligned} U_{\eta\to X}(\mathsf{mek}) &= \mathsf{Eval}(\mathsf{pp}, \mathsf{mek}, X), \\ U_{\eta\to Y}(\mathsf{mek}) &= \mathsf{Constrain}(\mathsf{pp}, \mathsf{mek}, Y), \text{ and} \\ U_{Y\to X}(\mathsf{ek}_Y) &= \mathsf{CEval}(\mathsf{pp}, \mathsf{ek}_Y, X). \end{aligned}$$

Then, the description of $U_{\eta\to X}$ is the concatenation of the description of $U_{Y\to X}$ and that of $U_{\eta\to Y}$.

The pseudo-randomness and the key simulation were also defined as the other properties. Please refer [29] for the details. Eventually, ABE utilizes the following fact of ccPRF.

Lemma 3 ([29])**.** *For any attribute $X \in \{0,1\}^\xi$, any access policy Y, any $(\mathsf{mek}, \mathsf{pp}) \leftarrow\!\!\$\ \mathsf{Pgen}(1^\lambda)$, and any "faked" key $\mathsf{ek}_Y \leftarrow\!\!\$\ \mathsf{KSim}(\mathsf{pp}, Y)$, the probability that $\mathsf{CEval}(\mathsf{pp}, \mathsf{ek}_Y, X) \in \{\bot, \mathsf{Eval}(\mathsf{pp}, \mathsf{mek}, X)\}$ is negligible when $Y(X) = 1$.*

A concrete ccPRF for a t-CNF Y is also given in [29].

3.3.3 Conversion of Evaluating Circuits into Lattices

[29] embedded ccPRF in ABE. In particular, ABE verifies whether or not the circuits $U_{\eta\to X}$, $U_{\eta\to Y}$ and $U_{Y\to X}$ are honestly evaluated. They also proposed a method to convert their evaluations into lattice forms by introducing two DPT algorithms EvalF and EvalFx in the following way. Let $\tilde{m} = n\lceil \log_2 q \rceil$, let $f : \{0,1\}^{\mathsf{in}} \to \{0,1\}^{\mathsf{out}}$ and $g : \{0,1\}^{\mathsf{out}} \to \{0,1\}^{\mathsf{out}'}$ be any boolean circuits with depth d, let $x \in \{0,1\}^{\mathsf{in}}$, and let $\boldsymbol{C} \in \mathbb{Z}_q^{n\times \tilde{m}\cdot \mathsf{in}}$. Given a pair $(f, \boldsymbol{C})$, EvalF returns a matrix $\boldsymbol{H} \in \mathbb{Z}_q^{\tilde{m}\cdot \mathsf{in}\times \tilde{m}\cdot \mathsf{out}}$, whereas given a tuple $(f, x, \boldsymbol{C})$, EvalFx returns a matrix $\hat{\boldsymbol{H}} \in \mathbb{Z}_q^{\tilde{m}\cdot \mathsf{in}\times \tilde{m}\cdot \mathsf{out}}$. The matrices $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ satisfy that

$$\begin{aligned} &\|\boldsymbol{H}\|, \left\|\hat{\boldsymbol{H}}\right\| \le (2\tilde{m})^d \text{ and} \\ &\left[\boldsymbol{C} - x \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}} = \boldsymbol{C}\boldsymbol{H} - f(x) \otimes \boldsymbol{G} \bmod q, \end{aligned}$$

where $\boldsymbol{G} = \begin{bmatrix} 1 & 2 & 4 & \cdots & 2^{\lceil \log q \rceil - 1} \end{bmatrix} \otimes \boldsymbol{I}_n \in \mathbb{Z}_q^{n\times \tilde{m}}$. When $\boldsymbol{H}_f \leftarrow \mathsf{EvalF}(f, \boldsymbol{C})$, $\boldsymbol{H}_g \leftarrow \mathsf{EvalF}(g, \boldsymbol{C}\boldsymbol{H}_f)$ and $\boldsymbol{H}_{g\circ f} \leftarrow \mathsf{EvalF}(g \circ f, \boldsymbol{C})$, it holds that $\boldsymbol{H}_f \boldsymbol{H}_g = \boldsymbol{H}_{g\circ f}$.

3.3.4 Construction of ABE

Fig. 4 is our CP-ABE part $\mathsf{ABE} = (\mathsf{APgen}, \mathsf{AKGen}, \mathsf{Enc}, \mathsf{Dec})$ with an auxiliary algorithm *dec* and Tab. 1 is the parameters.

We briefly explain the mechanism of the decryption. For any access policy $Y : \{0,1\}^\xi \to \{0,1\}$, any attribute $X \in \{0,1\}^\xi$ such that $Y(X) = 1$, and any message $\mu \in \{0,1\}^l$, we set

$$\begin{aligned} &((\boldsymbol{T}_{\boldsymbol{B}}, \eta), (\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{U}, \mathsf{pp})) \leftarrow\!\!\$\ \mathsf{APgen}(1^n, 1^{\tilde{m}}, q, \mathbb{U}), \\ &(X, \rho, \boldsymbol{K}) \leftarrow\!\!\$\ \mathsf{AKGen}((\boldsymbol{T}_{\boldsymbol{B}}, \eta), X), \\ &(Y, s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2) \leftarrow\!\!\$\ \mathsf{Enc}((\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{U}, \mathsf{pp}), Y, \mu) \text{ and} \\ &\tilde{\mu} \leftarrow \mathsf{Dec}((X, \rho, \boldsymbol{K}), (Y, s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2)). \end{aligned}$$

Recall that $\rho \leftarrow \mathsf{KSim}(\mathsf{pp}, Y)$ is not identical to $\rho' \leftarrow U_{Y\to X}(s_Y) = \mathsf{CEval}(\mathsf{pp}, s_Y, X)$ except the negligible error probability by Lemma 3, and hence we have $I_\rho(\rho') = 0$ for the circuit I_ρ generated

$\mathsf{APgen}(1^n, 1^{\tilde{m}}, q, \mathbb{U})$

$(\eta, \mathsf{pp}) \leftarrow\!\$\ \mathsf{CPgen}(1^\lambda)$; $(\boldsymbol{B}, \boldsymbol{T_B}) \leftarrow\!\$\ \mathsf{TrapGen}(1^n, 1^{\tilde{m}'}, q)$
$\boldsymbol{C} \leftarrow\!\$\ \mathbb{Z}_q^{n \times \tilde{m} \cdot \lambda}$; $\boldsymbol{U} \leftarrow\!\$\ \mathbb{Z}_q^{n \times l}$
$msk \leftarrow (\boldsymbol{T_B}, \eta)$; $\mathsf{pk} \leftarrow (\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{U}, \mathsf{pp})$
return (msk, pk)

$dec(\boldsymbol{u} = (\boldsymbol{u}[1], \ldots, \boldsymbol{u}[l]))$

$\mu[k] \leftarrow \begin{cases} 1 & |\boldsymbol{u}[k]| \le q/4 \\ 0 & |\boldsymbol{u}[k]| > q/4 \end{cases}$ **for** $k \in [1, l]$
return μ

$\mathsf{AKGen}(msk, X)$

$\boldsymbol{H}_{\eta \to X} \leftarrow \mathsf{EvalF}(U_{\eta \to X}, \boldsymbol{C})$; $\boldsymbol{C}_X \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta \to X}$
$\rho \leftarrow \mathsf{Eval}(\mathsf{pp}, \eta, X)$
Create a circuit $I_\rho : \{0,1\}^\lambda \to \{0,1\}$ **s.t.**
$I_\rho(\rho') \mapsto 1$ **if and only if** $\rho = \rho'$
$\boldsymbol{H}_\rho \leftarrow \mathsf{EvalF}(I_\rho, \boldsymbol{C}_X)$; $\boldsymbol{C}_{X,\rho} \leftarrow \boldsymbol{C}_X \boldsymbol{H}_\rho$
$\overline{\boldsymbol{B}} \leftarrow [\boldsymbol{B} \mid \boldsymbol{C}_{X,\rho}]$
$\boldsymbol{T}_{\overline{\boldsymbol{B}}} \leftarrow\!\$\ \mathsf{ExtBasis}(\overline{\boldsymbol{B}}, \boldsymbol{T_B})$
$\boldsymbol{K} \leftarrow\!\$\ \mathsf{SamplePre}(\overline{\boldsymbol{B}}, \boldsymbol{T}_{\overline{\boldsymbol{B}}}, \boldsymbol{U}, s)$
return $\mathsf{sk}_X \leftarrow (X, \rho, \boldsymbol{K})$

$\mathsf{Enc}(\mathsf{pk}, Y, \mu \in \{0,1\}^l)$

$s_Y \leftarrow\!\$\ \mathsf{KSim}(\mathsf{pp}, Y)$
$\boldsymbol{t} \leftarrow\!\$\ \chi^n$; $\boldsymbol{e}_0 \leftarrow\!\$\ \chi^{\tilde{m}'}$; $\boldsymbol{e}_1 \leftarrow\!\$\ (\chi')^{\tilde{m} \cdot \iota}$; $\boldsymbol{e}_2 \leftarrow\!\$\ \chi^l$
$\boldsymbol{H}_{\eta \to Y} \leftarrow \mathsf{EvalF}(U_{\eta \to Y}, \boldsymbol{C})$; $\boldsymbol{C}_Y \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta \to Y}$
$\boldsymbol{u}_0 \leftarrow \boldsymbol{B}^T \boldsymbol{t} + \boldsymbol{e}_0$; $\boldsymbol{u}_1 \leftarrow \left[\boldsymbol{C}_Y - s_Y \otimes \boldsymbol{G}\right]^T \boldsymbol{t} + \boldsymbol{e}_1$
$\boldsymbol{u}_2 \leftarrow \boldsymbol{U}^T \boldsymbol{t} + \boldsymbol{e}_2 + \mu \cdot \lceil q/2 \rfloor$
$\mathsf{cp} \leftarrow (s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2)$
return (Y, cp)

$\mathsf{Dec}(\mathsf{sk}_X, (Y, \mathsf{cp}))$

$\rho' \leftarrow U_{Y \to X}(s_Y)$
return $\perp$ **if** $Y(X) \neq 1 \vee \rho = \rho'$
$\boldsymbol{H}_{\eta \to Y} \leftarrow \mathsf{EvalF}(U_{\eta \to Y}, \boldsymbol{C})$; $\boldsymbol{C}_Y \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta \to Y}$
$\boldsymbol{H}_{\eta \to X} \leftarrow \mathsf{EvalF}(U_{\eta \to X}, \boldsymbol{C})$; $\boldsymbol{C}_X \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta \to X}$
$\hat{\boldsymbol{H}}_{s_Y \to \rho'} \leftarrow \mathsf{EvalFx}(U_{Y \to X}, s_Y, \boldsymbol{C}_Y)$
$\hat{\boldsymbol{H}}_{\rho, \rho'} \leftarrow \mathsf{EvalFx}(I_\rho, \rho', \boldsymbol{C}_X)$
$\hat{\boldsymbol{H}}_{s_Y \to \neq \rho'} \leftarrow \hat{\boldsymbol{H}}_{s_Y \to \rho'} \hat{\boldsymbol{H}}_{\rho, \rho'}$; $\overline{\boldsymbol{u}}_1 \leftarrow \hat{\boldsymbol{H}}^T_{s_Y \to \neq \rho'} \boldsymbol{u}_1$
return $dec\left(\boldsymbol{u}_2 - \boldsymbol{K}^T \begin{bmatrix} \boldsymbol{u}_0 \\ \overline{\boldsymbol{u}}_1 \end{bmatrix}\right)$

Figure 4: CP-ABE Part ABE (The plaintext space is $\{0,1\}^l$)

in AKGen. Since the concatenation of $U_{Y \to X}$ and $U_{\eta \to Y}$ is identical to $U_{\eta \to X}$ due to the gradual evaluation, we have $\boldsymbol{H}_{\eta \to Y} \boldsymbol{H}_{Y \to X} = \boldsymbol{H}_{\eta \to X}$, where $\boldsymbol{H}_{\eta \to Y}$ and $\boldsymbol{H}_{\eta \to X}$ are as in AKGen and $\boldsymbol{H}_{Y \to X}$ is defined by $\mathsf{EvalF}(U_{Y \to X}, \boldsymbol{C})$. It follows from the matrices $\boldsymbol{C}_X$ and $\boldsymbol{C}_Y$ generated in AKGen and Enc that

$$\boldsymbol{C}_Y \boldsymbol{H}_{Y \to X} = \boldsymbol{C}\boldsymbol{H}_{\eta \to Y} \boldsymbol{H}_{Y \to X} = \boldsymbol{C}\boldsymbol{H}_{\eta \to X} = \boldsymbol{C}_X.$$

It holds that

$$\left[\boldsymbol{C}_Y - s_Y \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}}_{s_Y \to \rho'} = \boldsymbol{C}_Y \boldsymbol{H}_{Y \to X} - U_{Y \to X}(s_Y) \otimes \boldsymbol{G} = \boldsymbol{C}_X - \rho' \otimes \boldsymbol{G} \tag{1}$$

$$\left[\boldsymbol{C}_X - \rho' \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}}_{\rho, \rho'} = \boldsymbol{C}_X \boldsymbol{H}_\rho - I_\rho(\rho') \otimes \boldsymbol{G} = \boldsymbol{C}_{X,\rho} \tag{2}$$

Putting together Eqs. (1) and (2), we have

$$\left[\boldsymbol{C}_Y - s_Y \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}}_{s_Y \to \neq \rho'} = \left[\boldsymbol{C}_Y - s_Y \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}}_{s_Y \to \rho'} \hat{\boldsymbol{H}}_{\rho, \rho'} = \left[\boldsymbol{C}_X - \rho' \otimes \boldsymbol{G}\right] \hat{\boldsymbol{H}}_{\rho, \rho'} = \boldsymbol{C}_{X,\rho}.$$

By letting $\overline{\boldsymbol{e}}_1 = \hat{\boldsymbol{H}}^T_{s_Y \to \neq \rho'} \boldsymbol{e}_1$, the following is obtained.

$$\overline{\boldsymbol{u}}_1 = \hat{\boldsymbol{H}}^T_{s_Y \to \neq \rho'} \boldsymbol{u}_1 = \boldsymbol{C}^T_{X,\rho} \boldsymbol{t} + \overline{\boldsymbol{e}}_1. \tag{3}$$

Observe from the algorithm AKGen that

$$\left[\boldsymbol{B} \mid \boldsymbol{C}_{X,\rho}\right] \boldsymbol{K} = \boldsymbol{U}. \tag{4}$$

Therefore, by setting $\boldsymbol{r} = \boldsymbol{e}_2 - \boldsymbol{K}^T \begin{bmatrix} \boldsymbol{e}_0 \\ \overline{\boldsymbol{e}}_1 \end{bmatrix}$, we have

$$\|\boldsymbol{r}\| \le q/4 \text{ and } \boldsymbol{u}_2 - \boldsymbol{K}^T \begin{bmatrix} \boldsymbol{u}_0 \\ \overline{\boldsymbol{u}}_1 \end{bmatrix} = \mu \lceil q/2 \rfloor + \boldsymbol{r}. \tag{5}$$

The detail can be seen in the proof of Lemma 4.1 in [29].

Table 1: Parameters of the proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$

N	# of group members	
ξ	length of attribute X	$\mathsf{poly}(\lambda)$
n	row	$\mathcal{O}(\lambda)$
q	modulus	$\tilde{\mathcal{O}}(\ell n^3)$
m	column for DS	$2n\lceil \log_2 q \rceil$
$\tilde{m}$	column of $\boldsymbol{C}$ for ABE	$n\lceil \log_2 q \rceil$
$\tilde{m}'$	column of $\boldsymbol{B}$ for ABE	$(n+1)\lceil \log_2 q \rceil + 2\lambda$
ℓ	length of τ of DS and length of N	$N = 2^\ell$
σ	s.d. of $\boldsymbol{d}$	$\Omega\big(\sqrt{n \log q} \log n\big)$
σ'	s.d. of $\boldsymbol{s}$	$\sqrt{(4\sqrt{2}\sigma m^{3/2})^2 + \sigma^2}$
β	infinite norm for DS	$\sigma\omega(\sqrt{m})$
κ	# of parallel of NIZK	$\omega(\log n)$
η	length of ρ in NIZK	the length of the witness given to P
l	plaintext length of ABE	$2m$
ϵ	efficiency ratio	$0 < \epsilon < 1$
d	depth of the circuits	$\mathsf{poly}(\lambda)$, $(2n^2)^{2d+4} \leq 2^{n^\epsilon}$
$\tilde{\sigma}$	s.d. for ABE	$\max\big\{\mathcal{O}\big(\sqrt{n \log q \log n}\big), \mathcal{O}\big(\lambda, (2\tilde{m})^{d+3}\big)\big\}$
B	B-bounded distribution χ	$q/B > 2^{n^\epsilon}$
B'	B'-bounded distribution χ'	$(\tilde{m} + \tilde{m}')\lambda B(2\tilde{m})^d$
ι	length of ek_Y	$\mathsf{poly}(\lambda)$

s.d.: standard deviation

Lemma 4 ([29]). ABE *is correct, and is* (T, ϵ)-IND-CPA *under* $\mathsf{LWE}_{n,q,\chi}$ *assumption, where* $T = \mathsf{poly}(\lambda)$ *and* $\epsilon = \mathsf{negl}(\lambda)$.

4 Proposed GSdT from Lattices

In this section, we propose a lattice-based GSdT $\mathsf{GSdT}_{\mathsf{lat}}$. A group signature by the proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$ is intuitively issued by encrypting the signer's identity vector $\boldsymbol{\zeta}_i$ generated in the joining protocol by using ABE, and then proving that the signer has been registered in the group and the ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$ of $\boldsymbol{\zeta}_i$ is validly generated. This proof π_E is generated by NIZK under an access policy Y with the target message μ as set in the label part lbl. The resulting group signature consists of $\Sigma = (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)$. All the algorithms of the proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$ are described in Fig. 5 and its parameters in Tab. 1. Note that the witnesses considered in GSig and Open are actually converted into the appropriate forms explained in Subsection 3.2.2. The details will be described below.

4.1 Construction

4.1.1 Joining Protocol

To join a group, a user $\mathcal{U}_i$ of index i generates his/her user key pair $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow\!\!\$\ \mathsf{UKG}(1^\lambda)$, which is a key pair of DS, in advance, runs the joining protocol with the issuer $\mathcal{I}$ who owns an issuer key ik which is a secret key of DS. Namely, $\mathcal{U}_i$ generates the binary representation $\boldsymbol{\zeta}_i \in \{0,1\}^{2m}$ of his/her identity vector $\boldsymbol{v}_i \leftarrow \boldsymbol{F}\boldsymbol{z}_i$ with a randomly chosen short vector $\boldsymbol{z}_i$, issues a signature $\Sigma_i \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{usk}_i, \boldsymbol{\zeta}_i)$, and then send $(\boldsymbol{v}_i, \Sigma_i)$ to $\mathcal{I}$. $\mathcal{I}$ issues a signature $cert_i \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{ik}, \boldsymbol{\zeta}_i)$, and then returns $cert_i$ to $\mathcal{U}_i$ with appending $\mathcal{U}_i$'s information $(\boldsymbol{v}_i, cert_i, \mathsf{upk}_i, \Sigma_i)$ into the registration table $\mathbf{reg}$. $\mathcal{U}_i$ stores $(\boldsymbol{z}_i, cert_i)$ as a group secret key gsk_i.

4.1.2 Group Signing and Verifying

To sign a message μ under an access policy Y by using gsk_i, $\mathcal{U}_i$ generates a ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}) \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}_{\mathcal{OM}}, Y, \boldsymbol{\zeta}_i)$, and then issues a proof π that $cert_i = (\tau_i, \boldsymbol{d}_i, \boldsymbol{s}_i)$ is indeed a signature of the

$\mathsf{GKG}(1^\lambda, \mathbb{U})$

$(\mathsf{ik}, \mathsf{pk}_\mathcal{I}) = (\boldsymbol{T}_{\boldsymbol{A}}, (\boldsymbol{A}, \{\boldsymbol{A}_t\}_{t\in[0,\ell]}, \boldsymbol{D}, \boldsymbol{D}_0, \boldsymbol{D}_1, \boldsymbol{u})) \leftarrow\!\$\ \mathsf{SKGen}(1^\lambda)$

$(\mathsf{omk}, \mathsf{pk}_{\mathcal{OM}}) = ((\boldsymbol{T}_{\boldsymbol{B}}, \eta), (\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{U}, \mathsf{pp})) \leftarrow\!\$\ \mathsf{APgen}(1^n, 1^{\tilde{m}}, q, \mathbb{U})$

$\boldsymbol{F} \leftarrow\!\$\ \mathbb{Z}_q^{4n\times 4m}$

return $(\mathsf{ik}, \mathsf{omk}, \mathsf{gpk} = (\mathsf{pk}_\mathcal{I}, \mathsf{pk}_{\mathcal{OM}}, \boldsymbol{F}))$

$\mathsf{OKG}(\mathsf{gpk}, \mathsf{omk}, j, X)$

return $\mathsf{ok}_j \leftarrow\!\$\ \mathsf{AKGen}(\mathsf{omk}, X)$

$\mathsf{UKG}(1^\lambda)$

return $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow\!\$\ \mathsf{SKGen}(1^\lambda)$

Joining Protocol

$\mathcal{U}_i(\mathsf{gpk}, i, \mathsf{upk}_i, \mathsf{usk}_i)$		$\mathcal{I}(\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk}_i, \mathbf{reg})$
$\boldsymbol{z}_i \leftarrow\!\$\ D_{\mathbb{Z}^{4m},\sigma}$; $\boldsymbol{v}_i \leftarrow \boldsymbol{F}\boldsymbol{z}_i$; $\boldsymbol{\zeta}_i \leftarrow \mathsf{bin}(\boldsymbol{v}_i)$		
$\Sigma_i \leftarrow\!\$\ \mathsf{Sig}(\mathsf{usk}_i, \boldsymbol{\zeta}_i)$	$\xrightarrow{(\boldsymbol{\zeta}_i, \Sigma_i)}$	**abort if** $\mathsf{Vf}(\mathsf{upk}_i, \boldsymbol{\zeta}_i, \Sigma_i) \neq 1$
		abort if $\exists(cert, i, upk, sig)$ **s.t.** $(\boldsymbol{\zeta}_i, cert, i, upk, sig) \in \mathbf{reg}$
		$cert_i = (\tau_i, \boldsymbol{d}_i, \boldsymbol{s}_i) \leftarrow\!\$\ \mathsf{Sig}(\mathsf{ik}, \boldsymbol{\zeta}_i)$
abort if $\mathsf{Vf}(\mathsf{pk}_\mathcal{I}, \boldsymbol{\zeta}_i, cert_i) \neq 1$	$\xleftarrow{cert_i}$	$\mathbf{reg} \leftarrow \mathbf{reg} \cup \{(\boldsymbol{\zeta}_i, cert_i, i, \mathsf{upk}_i, \Sigma_i)\}$
return $\mathsf{gsk}_i \leftarrow (\boldsymbol{z}_i, cert_i)$		

$\mathsf{GSig}(\mathsf{gpk}, (\boldsymbol{z}_i, cert_i), Y, \mu)$

$\boldsymbol{v}_i \leftarrow \boldsymbol{F}\boldsymbol{z}_i$; $\boldsymbol{\zeta}_i \leftarrow \mathsf{bin}(\boldsymbol{v}_i)$; $(\tau_i, \boldsymbol{d}_i, \boldsymbol{s}_i) \leftarrow cert_i$; $\begin{bmatrix}\boldsymbol{d}_{i,1}\\ \boldsymbol{d}_{i,2}\end{bmatrix} \leftarrow \boldsymbol{d}_i$

$(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}) \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}_{\mathcal{OM}}, Y, \boldsymbol{\zeta}_i)$; $(s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2) \leftarrow \mathsf{cp}_{\boldsymbol{\zeta}_i}$

$\mathsf{it}_E \leftarrow (\mathsf{pk}_\mathcal{I}, \boldsymbol{B}, \boldsymbol{C}_Y, \boldsymbol{F}, \boldsymbol{U}, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}))$

$\mathsf{wt}_E \leftarrow ((\boldsymbol{z}_i, \boldsymbol{\zeta}_i), cert_i, (\boldsymbol{t}, \boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}_2))$

$\pi_E \leftarrow\!\$\ P(\mathsf{it}_E, \mathsf{wt}_E, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}, \mu))$

return $\Sigma = (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))$

$\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)))$

$\boldsymbol{H}_{\eta\to Y} \leftarrow \mathsf{EvalF}(U_{\eta\to Y}, \boldsymbol{C})$

$\boldsymbol{C}_Y \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta\to Y}$

return $V(\mathsf{it}_E, \pi_E, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}, \mu))$

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}_j, \mathbf{reg}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)))$

return $(0, \perp)$ **if** $\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))) \neq 1$

$\boldsymbol{\zeta}_i \leftarrow \mathsf{Dec}(\mathsf{ok}_j, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}))$

find $(cert, i, \mathsf{upk}, \Sigma)$ **s.t.** $(\boldsymbol{\zeta}_i, cert, i, \mathsf{upk}, \Sigma_i) \in \mathbf{reg}$

return $(0, \perp)$ **if** $(\boldsymbol{\zeta}_i, cert, i, \mathsf{upk}, \Sigma_i) \notin \mathbf{reg}$

$\mathsf{it}_D \leftarrow (\boldsymbol{B}, \boldsymbol{C}_X, \boldsymbol{C}_{X,\rho}, \boldsymbol{U}, \rho', \hat{\boldsymbol{H}}_{s_Y\to\neq\rho'}, \overline{\boldsymbol{u}}_1, \boldsymbol{r}, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}, \boldsymbol{\zeta}_i))$

$\mathsf{wt}_D \leftarrow (\boldsymbol{K}, \hat{\boldsymbol{H}}_{\rho,\rho'}, \boldsymbol{H}_\rho)$; $\tau_D \leftarrow P(\mathsf{it}_D, \mathsf{wt}_D, (X, i))$

$\tau \leftarrow (cert, \Sigma_i, \boldsymbol{\zeta}_i, \tau_D, \boldsymbol{C}_{X,\rho}, \hat{\boldsymbol{H}}_{s_Y\to\neq\rho'}, \overline{\boldsymbol{u}}_1, \boldsymbol{r})$

return (i, τ)

$\mathsf{Judge}(\mathsf{gpk}, i, \mathsf{upk}_i, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)), \tau)$

return $\neg\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)))$ **if** $(i, \tau) = (0, \perp)$

$\rho' \leftarrow U_{Y\to X}(s_Y)$

$\boldsymbol{H}_{\eta\to X} \leftarrow \mathsf{EvalF}(U_{\eta\to X}, \boldsymbol{C})$

$\boldsymbol{H}_{\eta\to Y} \leftarrow \mathsf{EvalF}(U_{\eta\to Y}, \boldsymbol{C})$

$\boldsymbol{C}_X \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta\to X}$; $\boldsymbol{C}_Y \leftarrow \boldsymbol{C}\boldsymbol{H}_{\eta\to Y}$

$\hat{\boldsymbol{H}}_{s_Y\to\rho'} \leftarrow \mathsf{EvalFx}(U_{Y\to X}, s_Y, \boldsymbol{C}_Y)$

return 0 **if** Eq. (1) or Eq. (3) does not hold

return 1 **if** $V(\mathsf{it}_D, \tau_D, (X, i)) = 1 \wedge$
$\mathsf{Vf}(\mathsf{upk}_i, \boldsymbol{\zeta}_i, \Sigma_i) = 1 \wedge \mathsf{Vf}(\mathsf{pk}_\mathcal{I}, \boldsymbol{\zeta}_i, cert) = 1$

Figure 5: Proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$

Instance it_E

- the matrix $\boldsymbol{F} \in \mathbb{Z}_q^{4n \times 4m}$ for the user's identity vector
- the matrices $\boldsymbol{A}, \boldsymbol{A}_0, \{\boldsymbol{A}_t\}_{t=1}^{\ell} \in \mathbb{Z}_q^{n \times m}$, $\boldsymbol{D} \in \mathbb{Z}_q^{n \times n\lceil \log_2 q \rceil}$ and $\boldsymbol{D}_0 \in \mathbb{Z}_q^{2n \times 2m}$,$\boldsymbol{D}_1 \in \mathbb{Z}_q^{2n \times l}$ and the vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ for DS
- the matrices $\boldsymbol{B} \in \mathbb{Z}_q^{n \times \tilde{m}'}$, $\boldsymbol{C}_Y \in \mathbb{Z}_q^{n \times \tilde{m} \cdot \iota}$ and $\boldsymbol{U} \in \mathbb{Z}_q^{n \times l}$ for ABE
- the ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$ with $\mathsf{cp}_{\boldsymbol{\zeta}_i} = (s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2) \in \{0,1\}^{\iota} \times \mathbb{Z}_q^{\tilde{m}'} \times \mathbb{Z}_q^{\tilde{m} \cdot \iota} \times \mathbb{Z}_q^{l}$ of $\boldsymbol{\zeta}_i$

Witness wt_E

- $\mathcal{U}_i$'s identity secret $\boldsymbol{z}_i \in [\pm\beta]^{4m}$ for $\mathcal{U}_i$'s identity vector $\boldsymbol{v}_i$
- the binary representation $\boldsymbol{\zeta}_i \in \{0,1\}^{2m}$ of $\boldsymbol{v}_i$
- the signature $cert_i = (\tau_i, \boldsymbol{d}_{i,1}, \boldsymbol{d}_{i,2}, \boldsymbol{s}_i) \in \{0,1\}^{\ell} \times [\pm\beta]^m \times [\pm\beta]^m \times [\pm\beta]^l$ of $\boldsymbol{\zeta}_i$, where $\tau_i = \begin{bmatrix}\tau_i[1] & \cdots & \tau_i[\ell]\end{bmatrix}^T$
- the string $\boldsymbol{\gamma} \in \{0,1\}^{2n\lceil \log_2 q \rceil}$ appeared during generating $cert_i$
- the components $\boldsymbol{t} \in [\pm B]^n$, $\boldsymbol{e}_0 \in [\pm B]^{\tilde{m}'}$, $\boldsymbol{e}_1 \in [\pm B']^{\tilde{m} \cdot \iota}$ and $\boldsymbol{e}_2 \in [\pm B]^l$ appeared during generating $\mathsf{cp}_{\boldsymbol{\zeta}_i}$

These satisfy that

$$\boldsymbol{F}\boldsymbol{z}_i = \boldsymbol{J}_{4n \times 2m} \cdot \boldsymbol{\zeta}_i \bmod q$$

$$\boldsymbol{D}_0\boldsymbol{s} + \boldsymbol{D}_1 \cdot \boldsymbol{\zeta}_i = \boldsymbol{J}_{2n \times 2n\lceil \log_2 q \rceil} \cdot \boldsymbol{\gamma} \bmod q, \quad \boldsymbol{A} \cdot \boldsymbol{d}_1 + \boldsymbol{A}_{\boldsymbol{0}} \cdot \boldsymbol{d}_2 + \sum_{t=1}^{\ell} \boldsymbol{A}_t(\tau_i[t] \cdot \boldsymbol{d}_2) - \boldsymbol{D} \cdot \boldsymbol{\gamma} = \boldsymbol{u} \bmod q,$$

$$\boldsymbol{B}^T\boldsymbol{t} + \boldsymbol{e}_0 = \boldsymbol{u}_0 \bmod q, \left[\boldsymbol{C}_Y - s_Y \otimes \boldsymbol{G}\right]^T\boldsymbol{t} + \boldsymbol{e}_1 = \boldsymbol{u}_1 \bmod q, \text{ and } \boldsymbol{U}^T\boldsymbol{t} + \boldsymbol{e}_2 + (\lceil q/2 \rfloor \cdot \boldsymbol{I}_l)\boldsymbol{\zeta}_i = \boldsymbol{u}_2 \bmod q,$$

where $\boldsymbol{J}_{N \times N\lceil \log q \rceil} = \boldsymbol{I}_N \otimes \begin{bmatrix}1 & 2 & \cdots & 2^{\lceil \log q \rceil - 1}\end{bmatrix}$, and hence $\boldsymbol{v} = \boldsymbol{J}_{N \times N\lceil \log q \rceil} \cdot \mathsf{bin}(\boldsymbol{v})$ for any vector $\boldsymbol{v} \in \mathbb{Z}_q^N$.

Figure 6: Instance and Witness on Group Signing

binary representation $\boldsymbol{\zeta}_i$ of the $\mathcal{U}_i$'s identity vector $\boldsymbol{v}_i = \boldsymbol{F}\boldsymbol{z}_i$ signed by $\mathcal{I}$ and $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$ is indeed an encryption of $\boldsymbol{\zeta}_i$. More specifically, the instance and the witness are specified in Fig. 6. As mentioned in Subsection 3.2.2, the witness should be converted into the appropriate form. $\boldsymbol{z}_i \in [\pm\beta]^{4m}$, $\boldsymbol{s}, \boldsymbol{d}_1, \boldsymbol{d}_2 \in [\pm\beta]^m$, $\boldsymbol{t} \in [\pm B]^n$, $\boldsymbol{e}_0 \in [\pm B]^{\tilde{m}'}$, $\boldsymbol{e}_1 \in [\pm B']^{\tilde{m} \cdot \iota}$ and $\boldsymbol{e}_2 \in [\pm B]^l$ are converted into (Type 1), $\boldsymbol{\gamma} \in \{0,1\}^m$ and $\boldsymbol{\zeta}_i \in \{0,1\}^{2m}$ are done into (Type 2), and $(\boldsymbol{d}_2, \tau_i) \in [\pm\beta]^{2m} \times \{0,1\}^{\ell}$ is done into (Type 3). Then, the parameterized permutation corresponding to the converted witness is defined in the concatenation way.

4.1.3 Opening and Judging

The opener $\mathcal{OP}_j$ with the attribute X can trace a signer of a signature $(Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))$ if his/her attribute X satisfies the access policy Y. Namely, when $Y(X) = 1$, $\mathcal{OP}_j$ decrypts the ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$ with $\mathsf{cp}_{\boldsymbol{\zeta}_i} = (s_Y, \boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2)$ using Dec, and then finds the user index i for the decrypted result $\boldsymbol{\zeta}_i$ from the registration table $\mathbf{reg}$. To guarantee the non-frameability, $\mathcal{OP}_j$ also generates a proof τ_D that $\boldsymbol{\zeta}_i$ is correctly decrypted. The correct decryption implies that Eqs. (1)–(5) hold. Observe that whether or not Eq. (1) holds can be verified in public, whereas ρ in sk_X should be private, and hence the matrices $\boldsymbol{H}_\rho$ and $\hat{\boldsymbol{H}}_{\rho,\rho'}$ are difficult to be computed in the verification of τ_D during Judge. Instead, the knowledge of $\boldsymbol{H}_\rho$ and $\hat{\boldsymbol{H}}_{\rho,\rho'}$ are also proven simultaneously by presenting $\boldsymbol{C}_{X,\rho}$, $\hat{\boldsymbol{H}}_{s_Y \to \neq \rho'}$, $\overline{\boldsymbol{u}}_1$ and $\boldsymbol{r}$ during Dec as public. This implies that Eq. (3) can be verified in public. Therefore, the instance and the witness are specified in Fig. 7. In a similar manner to the signing, the representation of the witness $\boldsymbol{K}$, $\boldsymbol{H}_\rho$ and $\hat{\boldsymbol{H}}_{\rho,\rho'}$ is replaced with (Type 1). Moreover, these are matrices rather than vectors, and hence the proof is issued for each column of the converted matrices.

4.2 Security

We show the security of the proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$. These can be proven in the same way as [9, 3, 22].

Theorem 1 (Correctness)**.** $\mathsf{GSdT}_{\mathsf{lat}}$ *is* 0*-correct.*

Proof. Let $\mathcal{A}$ be an adversary for the correctness. As in $\mathsf{Exp}^{\mathrm{corr}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$, the user index i returned by $\mathcal{A}$ satisfies that $i \in HU$ and $\mathsf{gsk}[i] \neq \epsilon$. This implies that AddUO created an honest user secret key $\mathsf{gsk}[i] = (\boldsymbol{z}_i, cert_i)$ for i of a secret vector $\boldsymbol{z}_i$ and a signature $cert_i$ of $\boldsymbol{\zeta}_i = \mathsf{bin}(\boldsymbol{F}\boldsymbol{z}_i)$ from the issuer $\mathcal{I}$. Therefore, GSig can correctly issue a proof π_E of the knowledge of $\boldsymbol{z}_i$ that is signed by $\mathcal{I}$ and is correctly encrypted. Hence, GSig can generate a valid group signature $\Sigma = (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))$ with respect to (Y, μ) returned by $\mathcal{A}$. It follows from the completeness of NIZK shown in Lemma 2 that GVf can correctly verify the proof π_E. Therefore, GVf always returns 1.

On the other hand, for any index $j \in OS_Y$, the attributes X of the opener j satisfy $Y(X) = 1$ for the access structure Y returned by $\mathcal{A}$. The correctness of ABE shown in Lemma 4 and the correct secret key for X generated during AddOO imply that the ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$ is correctly decrypted to the valid identity vector $\boldsymbol{\zeta}_i$ that has been encrypted during GSig. Then, Open can find i's information $(\boldsymbol{\zeta}_i, cert, i, \mathbf{upk}[i], \Sigma_i)$ from the registration table $\mathbf{reg}$. Therefore, Open can issue a proof π_D which proves the correctness of the decryption of $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$, and hence Open returns i and a valid proof π_D. These also imply that Judge can confirm that GVf returns 1 and π_D is a valid proof by V. Moreover, since AddOO honestly generates signatures $cert$ and Σ_i of $\boldsymbol{\zeta}_i$, and therefore the correctness of DS shown in Lemma 1 implies that Vf always returns 1. These eventually mean that Judge always returns 1. Thus, the correctness of $\mathsf{GSdT}_{\mathsf{lat}}$ holds. □

Theorem 2 (Anonymity)**.** *Under the* $\mathsf{LWE}_{n,q,\chi}$ *assumption,* $\mathsf{GSdT}_{\mathsf{lat}}$ *is* $(T_{\mathrm{anom}}, \epsilon_{\mathrm{anom}})$*-anonymous in the random oracle model for a polynomial* T_{anom} *and a negligible function* ϵ_{anom}*.*

Proof. This is shown by the hybrid argument. Let $\mathcal{A}$ be an adversary violating the anonymity of $\mathsf{GSdT}_{\mathsf{lat}}$. The details are as follows.

Game_0 This game is identical to the experiment $\mathsf{Exp}^{\text{anom-0}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ of the anonymity of $\mathsf{GSdT}_{\mathsf{lat}}$ when $b = 0$. Here, all the oracles of the anonymity game are the same as in the definition. All of the hash values are given from the random oracle H. Then, we have

$$\Pr[\mathsf{Game}_0(\lambda) = 1] = \Pr\big[\mathsf{Exp}^{\text{anom-0}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\big]. \tag{6}$$

Game_1 This game proceeds in the same way as Game_0 except that the new state C is initialized as $\emptyset$ before running $\mathcal{A}$, the random oracle H is replaced with the simulator $\mathsf{Sim}_{\mathsf{H}}$ whose existence is guaranteed by the zero-knowledge of NIZK shown in Lemma 2, and ChaO-0 and OpenO are replaced with those as in Fig. 8. In the new ChaO-0, the proof π_E is generated by Sim_P for the zero-knowledge of NIZK, and the proof π_D is also generated by Sim_P. Observe that the difference between Game_0 and Game_1 is just the generation of the hash values and the proofs. Namely, all hash values and all proofs are given from the random oracle H and the prover algorithm P of NIZK in Game_0, whereas

Instance it_D

- the matrices $\boldsymbol{B} \in \mathbb{Z}_q^{n \times \tilde{m}'}$, $\boldsymbol{C}_X \in \mathbb{Z}_q^{n \times \tilde{m} \cdot \lambda}$, $\boldsymbol{U} \in \mathbb{Z}_q^{n \times l}$
- the string $\rho' \in \{0,1\}^\lambda$
- the components $\boldsymbol{C}_{X,\rho} \in \mathbb{Z}_q^{n \times \tilde{m}}$, $\hat{\boldsymbol{H}}_{s_Y \to \neq \rho'} \in \mathbb{Z}_q^{\tilde{m} \cdot \iota \times \tilde{m}}$, $\overline{\boldsymbol{u}}_1 \in \mathbb{Z}_q^{\tilde{m}}$, $\boldsymbol{r} \in [-q/4, q/4]^l$ computed during Dec
- the ciphertext $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i})$, and the decrypted result $\boldsymbol{\zeta}_i \in \{0,1\}^l$

Witness wt_D the following matrices:

$$\boldsymbol{K} \in [\pm\tilde{\sigma}\sqrt{\tilde{m} + \tilde{m}'}]^{(\tilde{m}+\tilde{m}') \times l}, \quad \boldsymbol{H}_\rho \in [\pm(2\tilde{m})^d]^{\tilde{m} \cdot \lambda \times \tilde{m}} \text{ and } \hat{\boldsymbol{H}}_{\rho,\rho'} \in [\pm(2\tilde{m})^d]^{\tilde{m} \cdot \iota \times \tilde{m}}$$

These satisfy that

$$[\boldsymbol{B} \mid \boldsymbol{C}_{X,\rho}]\, \boldsymbol{K} = \boldsymbol{U} \bmod q, \quad \boldsymbol{C}_X \boldsymbol{H}_\rho = \boldsymbol{C}_{X,\rho} \bmod q, \quad \hat{\boldsymbol{H}}_{s_Y \to \rho} \hat{\boldsymbol{H}}_{\rho,\rho'} = \hat{\boldsymbol{H}}_{s_Y \to \neq \rho'} \bmod q,$$

$$[\boldsymbol{C}_X - \rho' \otimes \boldsymbol{G}]\, \hat{\boldsymbol{H}}_{\rho,\rho'} = \boldsymbol{C}_{X,\rho} \bmod q, \text{ and } \boldsymbol{u}_2 - \boldsymbol{K}^T \begin{bmatrix} \boldsymbol{u}_0 \\ \overline{\boldsymbol{u}}_1 \end{bmatrix} = \boldsymbol{\zeta}_i \lceil q/2 \rfloor + \boldsymbol{r} \bmod q.$$

Figure 7: Instance and Witness on Opening

$\mathsf{ChaO}\text{-}b(i_0, i_1, \mu, Y)$

return $\perp$ **if**
  $\mathbf{gsk}[i_0] = \epsilon \vee \mathbf{gsk}[i_1] = \epsilon \vee \exists j \in HU \cup CU$ **s.t.**
  $Y(X) = 1 \wedge (X, \mathsf{ok}_0) = \mathbf{ok}[j]$
$(\boldsymbol{z}_{i_b}, cert_{i_b}) \leftarrow \mathbf{gsk}[i_b]$
$\boldsymbol{v}_{i_b} \leftarrow \boldsymbol{F}\boldsymbol{z}_{i_b}$; $\boldsymbol{\zeta}_{i_b} \leftarrow \mathsf{bin}(\boldsymbol{v}_{i_b})$
  // Game_0–Game_2, Game_5–Game_7
$\boldsymbol{\zeta}_{i_b} \leftarrow 0^{2m}$ // Game_3–Game_4
$(Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}) \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}_{\mathcal{OM}}, Y, \boldsymbol{\zeta}_{i_b})$
$C \leftarrow C \cup \left\{(Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}})\right\}$
$\mathsf{it}_E \leftarrow (\mathsf{pk}_{\mathcal{I}}, \boldsymbol{B}, \boldsymbol{C}_Y, \boldsymbol{F}, \boldsymbol{U}, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}))$
$\mathsf{wt}_E \leftarrow ((\boldsymbol{z}_i, \boldsymbol{\zeta}_i), cert_i, (\boldsymbol{t}, \boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}_2))$
  // Game_0,Game_7
$\pi_E \leftarrow\!\!\$\ P(\mathsf{it}_E, \mathsf{wt}_E, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}, \mu))$ // Game_0,Game_7
$\pi_E \leftarrow\!\!\$\ \mathsf{Sim}_P(\mathsf{it}_E, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}, \mu))$ // Game_1–Game_6
return $\Sigma \leftarrow (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}, \pi_E))$
$MS \leftarrow MS \cup \{(\mu, \Sigma)\}$

$\mathsf{OpenO}(j, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E)))$

1 : **return** $\perp$ **if** $(\mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))) \in MS$
2 : **return** $\perp$ **if** $\mathbf{ok}[j] = \perp$
3 : **return** $(0, \perp)$ **if** $\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_i}, \pi_E))) \neq 1$
4 : **abort if** $(Y, \mathsf{cp}_{\boldsymbol{\zeta}}) \in C$ // Game_2–Game_5
5 : $\boldsymbol{\zeta}_i \leftarrow \mathsf{Dec}(\mathsf{ok}_j, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}))$
6 : **find** $(cert, i, \mathsf{upk}, \Sigma)$ **s.t.** $(\boldsymbol{\zeta}_i, cert, i, \mathsf{upk}, \Sigma_i) \in \mathbf{reg}$
7 : **return** $(0, \perp)$ **if** $(\boldsymbol{\zeta}_i, cert, i, \mathsf{upk}, \Sigma_i) \notin \mathbf{reg}$
8 : $\mathsf{it}_D \leftarrow (\boldsymbol{B}, \boldsymbol{C}_X, \boldsymbol{C}_{X,\rho}, \boldsymbol{U}, \rho', \hat{\boldsymbol{H}}_{s_Y \to \neq \rho'}, \overline{\boldsymbol{u}}_1, \boldsymbol{r})$
9 : $\mathsf{wt}_D \leftarrow (\boldsymbol{K}, \hat{\boldsymbol{H}}_{\rho,\rho'}, \boldsymbol{H}_\rho)$ // Game_0,Game_7
10 : $\tau_D \leftarrow P(\mathsf{it}_D, \mathsf{wt}_D, (X, i))$ // Game_0,Game_7
11 : $\tau_D \leftarrow\!\!\$\ \mathsf{Sim}_P(\mathsf{it}_D, (X, i))$ // Game_1–Game_6
12 : $\tau \leftarrow (cert, \Sigma_i, \boldsymbol{\zeta}_i, \tau_D, \boldsymbol{C}_{X,\rho}, \hat{\boldsymbol{H}}_{s_Y \to \neq \rho'}, \overline{\boldsymbol{u}}_1, \boldsymbol{r})$
13 : **return** (i, τ)

Figure 8: $\mathsf{ChaO}\text{-}b$ and OpenO in Sequential Games for Anonymity, where // Game_* denotes that the corresponding line is performed only in Game_*

those are generated from $\mathsf{Sim}_{\mathsf{H}}$ and Sim_P in Game_1, respectively. It follows from Lemma 2 that

$$|\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}_0 = 1]| \leq \epsilon_{\mathsf{zk}}. \tag{7}$$

Game_2 This game proceeds in the same way as Game_1 except that the new abort condition is added as Line 4 in OpenO. Let C be the event that $(Y, \mathsf{cp}_{\boldsymbol{\zeta}_i}) \in C$ for a query to OpenO by $\mathcal{A}$. Observe that Game_2 coincides with Game_1 when the event C does not happen. This implies that

$$|\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_1 = 1]| \leq \Pr[\mathsf{C}]. \tag{8}$$

Game_3 This game proceeds in the same way as Game_2 except that $\boldsymbol{\zeta}_i$ is replaced with the zero vector 0^{2m} in $\mathsf{ChaO}\text{-}0$ instead of $\boldsymbol{v}_{i_b}$. The difference between Game_3 and Game_2 can be evaluated by the IND-CPA of ABE. Namely, we now construct an adversary $\mathcal{B}_{\mathsf{ABE}}$ for the IND-CPA of ABE as in Fig. 9.

In AddOO in Fig. 9, $\mathcal{B}_{\mathsf{ABE}}$ generates $\mathbf{ok}[j]$ by utilizes Ocorr, which is given to the IND-CPA adversary $\mathcal{B}_{\mathsf{ABE}}$ of ABE. This means that $\mathcal{B}_{\mathsf{ABE}}$ is prohibited from querying any access structure Y such that $Y(X) = 1$ for any attributes X given to AddOO. On the other hand, there is no possibility of winning the anonymity game. In fact, $\mathsf{ChaO}\text{-}b$ prohibits such an access structure as a $\mathsf{ChaO}\text{-}b$ query. Observe that the procedure of $\mathcal{B}_{\mathsf{ABE}}$ when $\mathsf{Och}\text{-}0$ is adopted coincides with that of Game_2, whereas the procedure of $\mathcal{B}_{\mathsf{ABE}}$ when $\mathsf{Och}\text{-}1$ is adopted coincides with that of Game_3. This implies that

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_2 = 1]| = \left|\Pr\left[\mathsf{Exp}^{\text{IND-CPA-0}}_{\mathsf{ABE},\mathcal{B}_{\mathsf{ABE}}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{IND-CPA-1}}_{\mathsf{ABE},\mathcal{B}_{\mathsf{ABE}}}(\lambda) = 1\right]\right| \leq \epsilon_{\mathsf{ABE}}. \tag{9}$$

Game_4 This game proceeds in the same way as Game_3 except that $\mathsf{ChaO}\text{-}0$ is replaced with $\mathsf{ChaO}\text{-}1$. Observe that there is no difference for the returned signature $\Sigma = (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}, \pi_E))$ between $\mathsf{ChaO}\text{-}0$

$\mathcal{B}^{\mathsf{Ocorr},\mathsf{Och}\text{-}d}_{\mathsf{ABE}}(\mathsf{pk}_{\mathsf{ABE}})$

$((\mathsf{pk}_{\mathcal{I}}, \mathsf{pk}'_{\mathcal{OM}}, \boldsymbol{F}), \mathsf{ik}, \mathsf{omk}) \leftarrow\!\!\$\ \mathsf{GKG}(1^{\lambda}, \mathbb{U})$; $\mathsf{gpk} \leftarrow (\mathsf{pk}_{\mathcal{I}}, \mathsf{pk}_{\mathsf{ABE}}, \boldsymbol{F})$
$HO \leftarrow \emptyset$; $HU \leftarrow \emptyset$; $CO \leftarrow \emptyset$; $CU \leftarrow \emptyset$
$\mathbf{upk} \leftarrow \emptyset$; $\mathbf{usk} \leftarrow \emptyset$; $\mathbf{gsk} \leftarrow \emptyset$; $\mathbf{ok} \leftarrow \emptyset$; $\mathbf{reg} \leftarrow \emptyset$; $MS \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Join}} \leftarrow \emptyset$; $\mathbf{st}_{\mathsf{Iss}} \leftarrow \emptyset$; $H \leftarrow \emptyset$
$C \leftarrow \emptyset$
return $\mathcal{A}^{\mathsf{ChaO}_b,\mathsf{AddOO},\mathsf{OpenO},\mathsf{StoUO},\mathsf{WRegO},\mathsf{USKO},\mathsf{CrptOO},\mathsf{CrptUO},\mathsf{H}}(\mathsf{gpk}, \mathsf{ik})$

$\mathsf{ChaO}\text{-}b(i_0, i_1, \mu, Y)$
return $\perp$ **if**
  $\mathbf{gsk}[i_0] = \epsilon \vee \mathbf{gsk}[i_1] = \epsilon \vee \exists j \in HU \cup CU$ **s.t.**
  $Y(X) = 1 \wedge (X, \mathsf{ok}_0) = \mathbf{ok}[j]$
$(\boldsymbol{z}_{i_b}, cert_{i_b}) \leftarrow \mathbf{gsk}[i_b]$
$\boldsymbol{v}_{i_b} \leftarrow \boldsymbol{F}\boldsymbol{z}_{i_b}$; $\boldsymbol{\zeta}_{i_b} \leftarrow \mathsf{bin}(\boldsymbol{v}_{i_b})$
$(Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}) \leftarrow\!\!\$\ \mathsf{Och}\text{-}d(Y, \boldsymbol{\zeta}_{i_b}, 0^{2m})$
$C \leftarrow C \cup \left\{(Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}})\right\}$
$\mathsf{it}_E \leftarrow (\mathsf{pk}_{\mathcal{I}}, \boldsymbol{B}, \boldsymbol{C}_Y, \boldsymbol{F}, \boldsymbol{U}, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}))$
$\pi_E \leftarrow\!\!\$\ \mathsf{Sim}_P(\mathsf{it}_E, (Y, \mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}, \mu))$
return $\boldsymbol{\Sigma} \leftarrow (Y, (\mathsf{cp}_{\boldsymbol{\zeta}_{i_b}}, \pi_E))$
$MS \leftarrow MS \cup \{(\mu, \boldsymbol{\Sigma})\}$

$\mathsf{AddOO}(j, X)$
return $\perp$ **if** $\exists j \in HO$
$HO \leftarrow HO \cup \{j\}$
$\mathbf{ok}[j] \leftarrow \mathsf{Ocorr}(X)$

Figure 9: IND-CPA Adversary $\mathcal{B}_{\mathsf{ABE}}$ for ABE from Game_1 and Game_2

in Game_3 and ChaO-1 in Game_4. Therefore, we have

$$\Pr[\mathsf{Game}_4 = 1] = \Pr[\mathsf{Game}_3 = 1]. \tag{10}$$

Game_5 This game proceeds in the same way as Game_4 except that $\boldsymbol{\zeta}_{i_b}$ is replaced with $\boldsymbol{v}_{i_b}$ in ChaO-1 instead of the zero vector 0^{2m}. In the same way as the evaluation of $|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_2 = 1]|$, we have

$$|\Pr[\mathsf{Game}_5 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \epsilon_{\mathsf{ABE}}. \tag{11}$$

Game_6 This game proceeds in the same way as Game_5 except that the abort condition at Line 4 is removed from OpenO. In the same way as the evaluation of $|\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_1 = 1]|$, we have

$$|\Pr[\mathsf{Game}_6 = 1] - \Pr[\mathsf{Game}_5 = 1]| \leq \Pr[\mathsf{C}]. \tag{12}$$

Game_7 This game proceeds in the same way as Game_6 except that the simulated random oracle $\mathsf{Sim}_{\mathsf{H}}$ and the simulated prover Sim_P are replaced with the ordinary random oracle H and the prover algorithm P. In the same way as the evaluation of $\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}_0 = 1]$, we have

$$|\Pr[\mathsf{Game}_7 = 1] - \Pr[\mathsf{Game}_6 = 1]| \leq \epsilon_{\mathsf{zk}}. \tag{13}$$

Observe that the procedure of Game_7 is identical to that of $\mathsf{Exp}^{\text{anom-1}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$. This implies that

$$\Pr[\mathsf{Game}_7 = 1] = \Pr\left[\mathsf{Exp}^{\text{anom-1}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\right]. \tag{14}$$

Putting these together Eqs. (6)–(14), we have

$$\epsilon_{\text{anom}} = \left|\Pr\left[\mathsf{Exp}^{\text{anom-0}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{anom-1}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\right]\right| \leq 2\epsilon_{\mathsf{ABE}} + 2\epsilon_{\mathsf{zk}} + 2\Pr[\mathsf{C}]. \tag{15}$$

Evaluation of Probability of C We remain in the evaluation of the probability of C. This is done by constructing an adversary $\mathcal{B}_{\mathsf{ss}}$ for the simulation soundness of NIZK with the black-box access to an adversary $\mathcal{A}$ violating Game_3. It should be noted that the event C is defined in Game_2 rather than Game_3. As shown above, the difference between Game_3 and Game_2 has been evaluated by connecting IND-CPA of ABE. Therefore, we here construct a reduction algorithm $\mathcal{B}_{\mathsf{ss}}$ breaking the simulation soundness of NIZK by the black-box access to the adversary $\mathcal{A}$ violating Game_3. $\mathcal{B}_{\mathsf{ss}}^{\mathsf{Sim}_{\mathsf{H}},\mathsf{OP}}$ runs in the following way:

(SS1) $\mathcal{B}_{\mathsf{ss}}$ runs Game_3 with the adversary $\mathcal{A}(\mathsf{gpk},\mathsf{ik})$ violating Game_3.

(SS2) $\mathcal{B}_{\mathsf{ss}}$ returns the tuple $((\mathsf{pk}_{\mathcal{I}},\boldsymbol{B},\boldsymbol{C}_Y,\boldsymbol{F},\boldsymbol{U},(Y,\mathsf{cp}_{\boldsymbol{\zeta}_i})),(Y,\mathsf{cp}_{\boldsymbol{\zeta}_i},\mu),\pi_E)$ for $(\mathsf{pk}_{\mathcal{I}},\boldsymbol{B},\boldsymbol{C}_Y,\boldsymbol{F},\boldsymbol{U})$ contained in the group public key gpk and the query $(\mu,Y,\mathsf{cp}_{\boldsymbol{\zeta}_i},\pi_E)$ given to OpenO as the final output if C happens.

When C happens, the query $(j,\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}_i},\pi_E)))$ satisfies the following conditions:

(1) $(\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}_i},\pi_E))) \notin MS$

(2) $\mathsf{ok}[j] \neq \perp$

(3) $\mathsf{GVf}(\mathsf{gpk},\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}_i},\pi_E))) = 1$

(4) $(Y,\mathsf{cp}_{\boldsymbol{\zeta}}) \in C$.

In particular, the condition (3) implies that $V((\mathsf{pk}_{\mathcal{I}},\boldsymbol{B},\boldsymbol{C}_Y,\boldsymbol{F},\boldsymbol{U}),(Y,\mathsf{cp}_{\boldsymbol{\zeta}_i},\mu),\pi_E)=1$. On the other hand, the condition (1) implies that π_E is not given from Sim_P. Moreover, the condition (4) implies that $(Y,\mathsf{cp}_{\boldsymbol{\zeta}})$ is the ciphertext issued in ChaO-b. This means that $(Y,\mathsf{cp}_{\boldsymbol{\zeta}})$ is the ciphertext of the zero vector 0^{2m}. It follows from the form of the instance described in Fig. 6 that $(\mathsf{pk}_{\mathcal{I}},\boldsymbol{B},\boldsymbol{C}_Y,\boldsymbol{F},\boldsymbol{U}) \notin L_{\mathsf{NIZK}}$. Therefore, π_E can be a forged proof for the simulation soundness of NIZK. Combining the success probability for the forged proof with the above note, it holds that

$$\Pr[\mathsf{C}] \leq \epsilon_{\mathsf{ABE}} + \epsilon_{\mathsf{ss}}. \tag{16}$$

By Eqs. (15) and (16), we have

$$\epsilon_{\mathrm{anom}} = \left|\Pr\left[\mathsf{Exp}^{\text{anom-0}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda)=1\right] - \Pr\left[\mathsf{Exp}^{\text{anom-1}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda)=1\right]\right| \leq 3\epsilon_{\mathsf{ABE}} + 2\epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{ss}}.$$

It follows from Lemmas 4 and 2 that ϵ_{anom} is evaluated as negligible under the $\mathsf{LWE}_{n,q,\chi}$ assumption. Hence $\mathsf{GSdT}_{\mathsf{lat}}$ is anonymous under the $\mathsf{LWE}_{n,q,\chi}$ assumption with the negligible function ϵ_{anom}. □

Theorem 3 (Traceability)**.** *Under the* $\mathsf{SIS}_{n,m,q,\beta}$ *assumption,* $\mathsf{GSdT}_{\mathsf{lat}}$ *is* $(T_{\mathrm{trac}},\epsilon_{\mathrm{trac}})$*-traceable for a polynomial* T_{trac} *and a negligible function* ϵ_{trac}*.*

Proof. Let $\mathcal{A}$ be an adversary violating the traceability of $\mathsf{GSdT}_{\mathsf{lat}}$. Lemma 1 implies that DS is EUF-CMA under the $\mathsf{SIS}_{n,m,q,\beta}$ assumption. Therefore, we show this theorem by constructing a reduction $\mathcal{R}_{\mathsf{DS}}$ breaking EUF-CMA of DS by the block-box access to an adversary $\mathcal{A}$ violating the traceability of $\mathsf{GSdT}_{\mathsf{lat}}$. Given a public key $\mathsf{pk}_{\mathsf{DS}}$ of DS to the EUF-CMA adversary $\mathcal{R}_{\mathsf{DS}}$,

(DS1) $\mathcal{R}_{\mathsf{DS}}$ generates omk and gpk in the same way as Fig. 5 except that $\mathsf{pk}_{\mathcal{I}} \leftarrow \mathsf{pk}_{\mathsf{DS}}$.

(DS2) $\mathcal{R}_{\mathsf{DS}}$ runs $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ with $\mathcal{A}(\mathsf{gpk},\mathsf{omk})$. Here, AddUO and StoIO are simulated as in Fig. 10 by utilizing the oracle Osig for the EUF-CMA game of DS.

(DS3) $\mathcal{R}_{\mathsf{DS}}$ aborts if $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ finally returns 0.

(DS4) For the tuple $(\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)))$ finally returned by $\mathcal{A}$, $\mathcal{R}_{\mathsf{DS}}$ decrypts $(Y,\mathsf{cp}_{\boldsymbol{\zeta}})$ to $\boldsymbol{\zeta}$ and then finds $\mathbf{upk}[i]$ and $\mathbf{gsk}[i]=(\boldsymbol{z},cert)$ from $\boldsymbol{\zeta}$ and the oracle's states.

(DS5) $\mathcal{R}_{\mathsf{DS}}$ aborts if $\mathsf{Vf}(\mathsf{pk}_{\mathcal{I}},\boldsymbol{\zeta},cert) \neq 1$.

(DS6) $\mathcal{R}_{\mathsf{DS}}$ returns $(\boldsymbol{\zeta},cert)$.

$\mathsf{AddUO}(i)$

return $\perp$ **if** $i \in HU \cup CU$
$(\mathbf{upk}[i], \mathbf{usk}[i]) \leftarrow\!\!\$\ \mathsf{UKG}(1^\lambda)$
$\boldsymbol{z}_i \leftarrow\!\!\$\ D_{\mathbb{Z}^{4m},\sigma}$; $\quad \boldsymbol{v}_i \leftarrow \boldsymbol{F}\boldsymbol{z}_i$; $\quad \boldsymbol{\zeta}_i \leftarrow \mathsf{bin}(\boldsymbol{v}_i)$
$\Sigma_i \leftarrow\!\!\$\ \mathsf{Sig}(\mathbf{usk}[i], \boldsymbol{\zeta}_i)$
abort if $\mathsf{Vf}(\mathbf{upk}[i], \boldsymbol{\zeta}_i, \Sigma_i) \neq 1$
abort if $\exists(cert, i, upk, sig)$
  s.t. $(\boldsymbol{\zeta}_i, cert, i, upk, sig) \in \mathbf{reg}$
$cert_i \leftarrow\!\!\$\ \mathsf{Osig}(\boldsymbol{\zeta}_i)$
abort if $\mathsf{Vf}(\mathsf{pk}_{\mathcal{I}}, \boldsymbol{\zeta}_i, cert_i) \neq 1$
$\mathbf{reg} \leftarrow \mathbf{reg} \cup \{(\boldsymbol{\zeta}_i, cert_i, i, \mathbf{upk}[i], \Sigma_i)\}$
$\mathbf{gsk}[i] \leftarrow (\boldsymbol{z}_i, cert_i)$
return $\mathbf{upk}[i]$
$HU \leftarrow HU \cup HU$; $\quad \mathbf{st}_{\mathsf{Join}} \leftarrow (\mathsf{gpk}, \mathbf{upk}[i], \mathbf{usk}[i])$

$\mathsf{StoIO}(i, M_{in})$

return $\perp$ **if** $i \notin CU$
$(\boldsymbol{\zeta}, \Sigma) \leftarrow M_{in}$
abort if $\mathsf{Vf}(\mathbf{upk}[i], \boldsymbol{\zeta}, \Sigma) \neq 1$
abort if $\exists(cert, i, upk, sig)$
  s.t. $(\boldsymbol{\zeta}, cert, i, upk, sig) \in \mathbf{reg}$
$cert_i \leftarrow\!\!\$\ \mathsf{Osig}(\boldsymbol{\zeta})$
abort if $\mathsf{Vf}(\mathsf{pk}_{\mathcal{I}}, \boldsymbol{\zeta}, cert_i) \neq 1$
$\mathbf{reg} \leftarrow \mathbf{reg} \cup \{(\boldsymbol{\zeta}, cert_i, i, \mathbf{upk}[i], \Sigma)\}$
return $M_{out} \leftarrow cert_i$

Figure 10: Oracles Simulated by EUF-CMAAdversary $\mathcal{R}_{\mathsf{DS}}^{\mathsf{Osig}}$ of DS

In a similar manner to Theorem 2, the difference of AddUO and StoIO from the original experiment $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ is that $cert_i$ is generated via the oracle Osig. Osig can be used, because $\mathcal{R}_{\mathsf{DS}}$ is now the EUF-CMA adversary. Therefore, the process (DS2) is equivalent to that of $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$.

We now evaluate the abort probability at the process (DS5). The process (DS3) guarantees that $\mathsf{GVf}(\mathsf{gpk}, \mu, (Y, (\mathsf{cp}_{\boldsymbol{\zeta}}, \pi_E))) = 1$. More specifically, the verifier V judges that π_E is a valid proof during Vf of $\mathsf{GSdT}_{\mathsf{lat}}$ in Fig. 5. Recall that π_E would prove that $\mathsf{Vf}(\mathsf{pk}_{\mathcal{I}}, \boldsymbol{\zeta}, cert) = 1$ as described in Fig. 6. This means that V judges that π_E is valid even when $\mathsf{Vf}(\mathsf{pk}_{\mathcal{I}}, \boldsymbol{\zeta}, cert) \neq 1$ if the abort happens. It follows from the soundness of NIZK explained in Lemma 2 that the abort probability is bounded by ϵ_s.

Since Open returns 0 to win the tracing game, it would follow that $\boldsymbol{\zeta}$ is not queried to the signing oracle. Therefore, $\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{DS},\mathcal{R}_{\mathsf{DS}}}$ returns 1 by the pair $(\boldsymbol{\zeta}, cert)$ if $\mathsf{Exp}^{\mathrm{trac}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ returns 1. More precisely, we have

$$\epsilon_{\mathrm{trac}} \leq \epsilon_{\mathsf{DS}} + \epsilon_s.$$

It follows from Lemmas 1 and 2 that ϵ_{trac} can be evaluated as negligible under the $\mathsf{SIS}_{n,m,q,\beta}$ assumption. Thus, $\mathsf{GSdT}_{\mathsf{lat}}$ is traceable under the $\mathsf{SIS}_{n,m,q,\beta}$ assumption with the negligible function ϵ_{trac}. □

Theorem 4 (Non-frameability)**.** *Under the* $\mathsf{SIS}_{4n,4m,q,4\sigma\sqrt{m}}$ *assumption,* $\mathsf{GSdT}_{\mathsf{lat}}$ *is* $(T_{\mathrm{nf}}, \epsilon_{\mathrm{nf}})$*-non-frameable in the random oracle model for a polynomial* T_{nf} *and a negligible function* ϵ_{nf}*.*

Proof. Let $\mathcal{A}$ be an adversary violating the non-frameability of $\mathsf{GSdT}_{\mathsf{lat}}$. We show this theorem by utilizing the weak simulation extractability of NIZK shown in Lemma 2. We first change the original experiment $\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ for the non-frameability in the random oracle model into the new experiment $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ in a sense that the random oracle H and the prover algorithm P appeared in $\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ are replaced with the simulated ones Sim_P and Sim_H in the same way as the game change from Game_0 to Game_1 in Theorem 2. Therefore, the difference can be evaluated as

$$\left|\Pr\left[\mathsf{Exp}^{\mathrm{nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda) = 1\right]\right| \leq \epsilon_{\mathsf{zk}}. \tag{17}$$

To apply the weak simulation extractability of NIZK to $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$, we construct an algorithm $\mathcal{B}_{\mathrm{nf}}$ in the following way: On a given matrix $\boldsymbol{F} \leftarrow\!\!\$\ \mathbb{Z}_q^{4n\times 4m}$,

(NF1) $\mathcal{B}_{\mathrm{nf}}$ runs $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ with $\mathcal{A}(\mathsf{gpk},\mathsf{ik},\mathsf{omk})$, where $\boldsymbol{F}$ of gpk is replaced with the given one.

(NF2) When $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ returns 1 with the final output $(\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)),i,\tau)$ of $\mathcal{A}$, $\mathcal{B}_{\mathrm{nf}}$ retrieves the instance $\mathsf{it}_E=(\mathsf{pk}_{\mathcal{I}},\boldsymbol{B},\boldsymbol{C}_Y,\boldsymbol{F},\boldsymbol{U},(Y,\mathsf{cp}_{\boldsymbol{\zeta}}))$ of π_E from gpk and the signature $(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E))$ generated during $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$. Note that $\boldsymbol{C}_Y$ can be retrieved by $\boldsymbol{C}_Y=\boldsymbol{C}H_{\eta\to Y}$ for $\boldsymbol{C}$ in gpk and $H_{\eta\to Y}\leftarrow\mathsf{EvalF}(U_{\eta\to Y},\boldsymbol{C})$. Then, $\mathcal{B}_{\mathrm{nf}}$ returns $(\mathsf{it}_E,(Y,\mathsf{cp}_{\boldsymbol{\zeta}},\mu),\pi_E)$.

When $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ returns 1, we have $\mathsf{GVf}(\mathsf{gpk},\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)))=1$, namely $V(\mathsf{it}_E,(Y,\mathsf{cp}_{\boldsymbol{\zeta}},\mu),\pi_E)=1$. Therefore, the probability acc by $\mathcal{B}_{\mathrm{nf}}$ for the weak simulation extractability of NIZK is evaluated as

$$acc=\Pr\Big[\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda)=1\Big]. \tag{18}$$

It follows from Lemma 2 and the definition of the weak simulation extractability that there exist an extraction algorithm $\mathsf{Ext}=(\mathsf{Ext}_1,\mathsf{Ext}_2)$, a constant d and a polynomial p such that

$$ext\geq\frac{1}{p}(acc-\nu_{\mathsf{Ext}})^d. \tag{19}$$

The definition of ext implies that Ext can extract a witness $\mathsf{wt}_E=((\hat{\boldsymbol{z}},\hat{\boldsymbol{\zeta}}),\hat{cert},(\hat{\boldsymbol{t}},\hat{\boldsymbol{e}}_0,\hat{\boldsymbol{e}}_1,\hat{\boldsymbol{e}}_2))$ of the instance it_E with the probability ext.

Utilizing Ext, we now construct an algorithm $\mathcal{R}_{\mathsf{SIS}}$ solving the SIS problem. On a given matrix $\boldsymbol{F}\leftarrow\!\!\$\ \mathbb{Z}_q^{4n\times 4m}$,

(SE1) $\mathcal{R}_{\mathsf{SIS}}$ runs $(st,\mathsf{it}_E,(Y,\mathsf{cp}_{\boldsymbol{\zeta}},\mu),\pi_E)\leftarrow\!\!\$\ \mathsf{Ext}_1(\boldsymbol{F})$ with the adversary $\mathcal{B}_{\mathrm{nf}}^{\mathsf{Sim}_{\mathsf{H}},\mathsf{Sim}_P}(\boldsymbol{F})$ defined above. Here we let $(\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)),i,\tau)$ with $\tau\leftarrow(cert,\Sigma_i,\boldsymbol{\zeta}_i,\tau_D,\boldsymbol{C}_{X,\rho},\hat{\boldsymbol{H}}_{s_Y\to\neq\rho'},\overline{\boldsymbol{u}}_1,\boldsymbol{r})$ be the final output of $\mathcal{A}$ during running $\mathcal{B}_{\mathrm{nf}}$.

(SE2) $\mathcal{R}_{\mathsf{SIS}}$ also runs $\mathsf{wt}_E=((\hat{\boldsymbol{z}},\hat{\boldsymbol{\zeta}}),\hat{cert},(\hat{\boldsymbol{t}},\hat{\boldsymbol{e}}_0,\hat{\boldsymbol{e}}_1,\hat{\boldsymbol{e}}_2))\leftarrow\mathsf{Ext}_2(st,\mathsf{it}_E,(Y,\mathsf{cp}_{\boldsymbol{\zeta}},\mu),\pi_E)$.

(SE3) $\mathcal{R}_{\mathsf{SIS}}$ retrieves $(\boldsymbol{z}_i,cert_i)\leftarrow\mathbf{gsk}[i]$ and $\boldsymbol{\zeta}_i$ from τ returned by $\mathcal{A}$.

(SE4) $\mathcal{R}_{\mathsf{SIS}}$ aborts if $\boldsymbol{z}_i=\hat{\boldsymbol{z}}\vee\boldsymbol{\zeta}_i\neq\hat{\boldsymbol{\zeta}}$.

(SE5) $\mathcal{R}_{\mathsf{SIS}}$ returns $\boldsymbol{z}_i-\hat{\boldsymbol{z}}$.

We now evaluate the abort probability at the process (SE4). We first show that $\boldsymbol{\zeta}_i=\hat{\boldsymbol{\zeta}}$ always holds. It follows from $(\mathsf{it}_E,\mathsf{wt}_E)\in R_{\mathsf{NIZK}}$ that $(Y,\mathsf{cp}_{\boldsymbol{\zeta}})$ contained in both the instance it_E and the label $(Y,\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)$ is a ciphertext of $\hat{\boldsymbol{\zeta}}$. On the other hand, since $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}$ returns 1, $\mathsf{Judge}(\mathsf{gpk},i,\mathbf{upk}[i],\mu,(Y,(\mathsf{cp}_{\boldsymbol{\zeta}},\pi_E)),\tau)$ also returns 1. In particular, Judge has verify that π_D is a valid proof as in Fig. 5. More precisely, the valid proof π_D guarantees for the open identity vector $\boldsymbol{\zeta}_i$ is indeed the decryption of $(Y,\mathsf{cp}_{\boldsymbol{\zeta}})$. The correctness of ABE shown in Lemma 4 implies that $\boldsymbol{\zeta}_i=\hat{\boldsymbol{\zeta}}$.

We next show the probability of $\boldsymbol{z}_i\neq\hat{\boldsymbol{z}}$. Since $\boldsymbol{\zeta}_i=\hat{\boldsymbol{\zeta}}$, it holds that $\boldsymbol{v}=\boldsymbol{F}\boldsymbol{z}_i=\boldsymbol{F}\hat{\boldsymbol{z}}$, where $\boldsymbol{\zeta}_i=\hat{\boldsymbol{\zeta}}=\mathsf{bin}(\boldsymbol{v})$. On the other hand, we can observe that $\mathcal{A}$ has no chance of knowing $\boldsymbol{z}_i$. In fact, the only chance to know $\boldsymbol{z}_i$ is obtaining $\mathbf{gsk}[i]=(\boldsymbol{z}_i,cert)$ by making a query the index i to the user secret key oracle USKO. However, such a query leads that $\mathsf{Exp}^{\mathsf{Sim\text{-}nf}}_{\mathsf{GSdT}_{\mathsf{lat}},\mathcal{A},\mathbb{U}}(\lambda)$ returns 0. Therefore, the statistical witness indistinguishability of NIZK shown in Lemma 2 and the min-entropy of $\boldsymbol{z}_i\leftarrow\!\!\$\ D_{\mathbb{Z}^{4m},\sigma}$ guarantee that $\boldsymbol{z}_i\neq\hat{\boldsymbol{z}}$ with probability $1-\epsilon_{\mathsf{wi}}$ as discussed in [23, Section C.2]. Thus, the abort probability at (SE4) can be evaluated as ϵ_{wi}.

If $\mathcal{R}_{\mathsf{SIS}}$ does not abort at (SE4), it holds that $\boldsymbol{F}(\boldsymbol{z}_i-\hat{\boldsymbol{z}})=\mathbf{0}$. As mentioned in Section 2.1, for each $\boldsymbol{z}\in\{\boldsymbol{z}_i,\hat{\boldsymbol{z}}\}$, we have $\|\boldsymbol{z}\|\leq\sigma\sqrt{4m}$ with at least probability $1-2^{\Omega(4m)}$. This implies that $\|\boldsymbol{z}_i-\hat{\boldsymbol{z}}\|\leq 4\sigma\sqrt{m}$ with probability at least $1-2^{\Omega(4m)+1}$.

For the algorithm $\mathcal{R}_{\mathsf{SIS}}$, we have

$$\Pr\big[\boldsymbol{F}(\boldsymbol{z}_i-\hat{\boldsymbol{z}})=\mathbf{0}\wedge\|\boldsymbol{z}_i-\hat{\boldsymbol{z}}\|\leq 4\sigma\sqrt{m}\big]\geq(1-\epsilon_{\mathsf{wi}})(1-2^{\Omega(4m)+1})ext=ext-\epsilon_{\mathsf{wi}}-2^{\Omega(4m)+1}. \tag{20}$$

Table 2: Comparisons among our proposed GSdT $\mathsf{GSdT}_{\mathsf{lat}}$, naive construction by [3] and GSdT from symmetric-key primitives by [5]

	[3]	[5]	$\mathsf{GSdT}_{\mathsf{lat}}$ [ours]
$\lvert\mathsf{ik}\rvert$	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$	—	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$
$\lvert\mathsf{omk}\rvert$	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$	—	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$
$\lvert\mathsf{ok}\rvert$	$\xi+\mathcal{O}\Big(d\ell\lambda^3\lvert\lambda\rvert^4\Big)$	$\mathcal{O}(N\lambda\lvert X\rvert)$	$\xi+\mathcal{O}\Big(d\lambda^2\lvert\lambda\rvert^3\Big)$
$\lvert\mathsf{gpk}\rvert$	$\mathcal{O}\Big(\ell\lambda^3\lvert\lambda\rvert^3\Big)$	$\mathcal{O}(N\lambda\lvert\mathbb{U}\rvert)$	$\mathcal{O}\Big(\ell\lambda^2\lvert\lambda\rvert^2\Big)$
$\lvert\mathsf{usk}\rvert$	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$	—	$\mathcal{O}\Big(\lambda^2\lvert\lambda\rvert^3\Big)$
$\lvert\mathsf{upk}\rvert$	$\mathcal{O}\Big(\ell\lambda^3\lvert\lambda\rvert^3\Big)$	—	$\mathcal{O}\Big(\ell\lambda^2\lvert\lambda\rvert^2\Big)$
$\lvert\mathsf{gsk}\rvert$	$\mathcal{O}\Big(\ell\lambda^2\lvert\lambda\rvert^2\Big)$	$\mathcal{O}(\lambda\lvert\mathbb{U}\rvert)$	$\mathcal{O}\Big(\ell+\lambda\lvert\lambda\rvert^2\Big)$
$\lvert\Sigma\rvert$	$\ell_Y+\mathcal{O}\Big((\ell\lambda\lvert\lambda\rvert+d\iota)\lambda\lvert\lambda\rvert^3\Big)$	$\mathcal{O}((\ell+\lvert\ell_Y\rvert)\lambda)$	$\ell_Y+\mathcal{O}\Big((\ell+d\iota)\lambda\lvert\lambda\rvert^2\Big)$
$\lvert\tau\rvert$	$\mathcal{O}\Big(d(\ell\lambda\lvert\lambda\rvert+\iota)\lambda^2\lvert\lambda\rvert^4\Big)$	—	$\mathcal{O}\Big(\ell+d\iota\lambda^2\lvert\lambda\rvert^4\Big)$
$\lvert\mathsf{id}\rvert$	$\mathcal{O}\Big(\ell\lambda^2\lvert\lambda\rvert^2\Big)$	$\mathcal{O}(\ell)$	$\mathcal{O}(\lambda\lvert\lambda\rvert)$
$\lvert\mathsf{wt}_E\rvert$	$\mathcal{O}\Big((\ell\lambda\lvert\lambda\rvert+d\iota)\lambda\lvert\lambda\rvert^2\Big)$	$\mathcal{O}((\ell+\lvert\ell_Y\rvert)\lambda)$	$\mathcal{O}\Big((\ell+d\iota)\lambda\lvert\lambda\rvert^2\Big)$
$\lvert\mathsf{wt}_D\rvert$	$\mathcal{O}\Big(d(\ell\lambda\lvert\lambda\rvert+\iota)\lambda^2\lvert\lambda\rvert^3\Big)$	—	$\mathcal{O}\Big(d\iota\lambda^2\lvert\lambda\rvert^3\Big)$
Joining	1 round $+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{RM}}+\mathcal{O}\Big(\ell\lambda^3\lvert\lambda\rvert^2\Big)T^{\mathsf{BM}}$ $+2(T^{\mathsf{ExtBasis}}_{n,2m,q}+T^{\mathsf{SamplePre}}_{n,2m,q,\sigma})+T^{\mathsf{TrapGen}}_{n,m,q}$	—	1 round $+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{RM}}+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{BM}}$ $+2(T^{\mathsf{ExtBasis}}_{n,2m,q}+T^{\mathsf{SamplePre}}_{n,2m,q,\sigma})$
GSig	$\mathcal{O}\Big((\ell\lambda\lvert\lambda\rvert+d\iota)(\ell\lambda\lvert\lambda\rvert+\iota)\lambda^4\lvert\lambda\rvert^4\Big)T^{\mathsf{RM}}$ $+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{BM}}+T^{\mathsf{KSim}}+T^{\mathsf{EvalF}}_{\xi,\iota}$ $+T^{\mathsf{ExtBasis}}_{n,2m,q}+T^{\mathsf{SamplePre}}_{n,2m,q,\sigma}$	— $\mathcal{O}\big(N^2\ell_Y^2+\mathsf{Mn}\big)T^{\mathsf{sym}}$	$\mathcal{O}\Big((\ell+d\iota)(\ell+\iota\lvert\lambda\rvert)\lambda\lvert\lambda\rvert^3\Big)T^{\mathsf{RM}}$ $+T^{\mathsf{KSim}}+T^{\mathsf{EvalF}}_{\xi,\iota}$
GVf	$\mathcal{O}\Big((\ell\lambda\lvert\lambda\rvert+d\iota)(\ell\lambda\lvert\lambda\rvert+\iota)\lambda^4\lvert\lambda\rvert^4\Big)T^{\mathsf{RM}}$ $+T^{\mathsf{EvalF}}_{\xi,\iota}$	$\mathcal{O}\big(N^2\ell_Y^2+\mathsf{Mn}\big)T^{\mathsf{sym}}$	$\mathcal{O}\Big((\ell+d\iota)(\ell+\iota\lvert\lambda\rvert)\lambda^2\lvert\lambda\rvert^3\Big)T^{\mathsf{RM}}$ $+T^{\mathsf{EvalF}}_{\xi,\iota}$
Open	$\mathcal{O}\Big(d(\ell\lambda\lvert\lambda\rvert+\iota)^2\lambda^3\lvert\lambda\rvert^5\Big)T^{\mathsf{RM}}$ $+2T^{\mathsf{EvalF}}_{\xi,\iota}+T^{\mathsf{EvalF}}_{\xi,\lambda}+T^{\mathsf{EvalFx}}_{\iota,\lambda}+T^{\mathsf{EvalFx}}_{\lambda,1}$	$\mathcal{O}\big(N^2\ell_Y^2+\mathsf{Mn}\big)T^{\mathsf{sym}}$	$\mathcal{O}\Big((\ell+d\iota)(\ell+\iota\lvert\lambda\rvert)\lambda^3\lvert\lambda\rvert^4\Big)T^{\mathsf{RM}}$ $+2T^{\mathsf{EvalF}}_{\xi,\iota}+T^{\mathsf{EvalF}}_{\xi,\lambda}+T^{\mathsf{EvalFx}}_{\iota,\lambda}+T^{\mathsf{EvalFx}}_{\lambda,1}$
Judge	$\mathcal{O}\Big(d(\ell\lambda\lvert\lambda\rvert+\iota)^2\lambda^3\lvert\lambda\rvert^5\Big)T^{\mathsf{RM}}$ $+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{BM}}$ $+2T^{\mathsf{EvalF}}_{\xi,\iota}+T^{\mathsf{EvalF}}_{\xi,\lambda}+T^{\mathsf{EvalFx}}_{\iota,\lambda}$	—	$\mathcal{O}\Big((\ell+d\iota)(\ell+\iota\lvert\lambda\rvert)\lambda^3\lvert\lambda\rvert^4\Big)T^{\mathsf{RM}}$ $+\mathcal{O}(\lambda^2\lvert\lambda\rvert)T^{\mathsf{BM}}$ $+2T^{\mathsf{EvalF}}_{\xi,\iota}+T^{\mathsf{EvalF}}_{\xi,\lambda}+T^{\mathsf{EvalFx}}_{\iota,\lambda}$
Y	t-CNF	all-and	t-CNF
Model	Partially dynamic	Static	Partially dynamic

Putting together with Eqs. (17)–(20), we have

$$\epsilon_{\mathrm{nf}} \leq \sqrt[d]{p(\epsilon_{\mathsf{SIS}}+\epsilon_{\mathsf{wi}}+2^{\Omega(4m)+1})}+\epsilon_{\mathsf{zk}}+\nu_{\mathsf{Ext}}.$$

It follows from the $\mathsf{SIS}_{4m,4n,q,4\sigma\sqrt{m}}$ assumption and Lemma 2 that the left-hand side of the inequality is negligible, and hence ϵ_{nf} is negligible. Thus, $\mathsf{GSdT}_{\mathsf{lat}}$ is non-frameable under $\mathsf{SIS}_{4m,4n,q,4\sigma\sqrt{m}}$ assumption with the negligible function ϵ_{nf}. □

5 Comparison

We compare the efficiency of $\mathsf{GSdT}_{\mathsf{lat}}$ with the naive construction, which is yielded from the generic construction of a GSdT by [3], and the GSdT from symmetric-key primitives by [5]. The generic construction by [3] employs a signature scheme, a CP-ABE and a simulation-sound NIZK as building blocks. Thus we apply our sub-algorithms DS, ABE and NIZK into the generic construction for comparison. The results are summarized in Tab. 2, where ℓ_Y denotes the length of the representation of the access policy Y, $|\lambda| = \log\lambda$, $|\ell_Y| = \log\ell_Y$, $|X|$ is the number of attributes involved in X,

$|\mathbb{U}|$ is the number of attributes in the attribute universe, and (M, n) are the parameters related to the MPC-in-the-head paradigm (see [5] for the details). Note that several items of [5] are written as "—". This means that the corresponding items do not exist, since the construction model of [5] differs from ours. More specifically, they employed the static model [8], while we employ the (partially) dynamic model [9]. For computational efficiency for $\mathsf{GSdT}_{\mathsf{lat}}$ and [3], we only pick up the following notable computational costs because they are dominant in the overall computational costs. T^{RM} is the time for the multiplication over $\mathbb{Z}_q$. T^{BM} is the time for the multiplication of a bit and an element in $\mathbb{Z}_q$. $T^{\mathsf{Alg}}_{\mathrm{par}}$ denotes the running time of an auxiliary algorithm Alg with parameters par for an input. Note that we also take into account $T^{\mathsf{Alg}}_{\mathrm{par}}$ performed in the sub-algorithms DS, NIZK and ABE. T^{sym} denotes the upper bound of the running times of the symmetric-key primitives involving the hashing, the commitment, the encryption, and the decryption.

We first discuss the comparison between $\mathsf{GSdT}_{\mathsf{lat}}$ and [3]. The sizes of a group public key gpk, an opening key ok, a group secret key gsk and a signature Σ of ours are significantly shorter than those of [3] as in Tab. 2. The main reason is that an encrypted identity ζ of ours during GSig is remarkably shorter than that of [3]. Such an identity is denoted by id in Tab. 2. Recall that the encrypted identity ζ of ours is just $2m$ length, namely a $\mathcal{O}(\lambda|\lambda|)$ length string as shown in Subsect. 4.1.1. On the other hand, the encrypted tuple id of [3] consists of an index $i \in [1, N]$, a public key for DS and two signatures generated by DS, and hence $|\mathsf{id}|$ of theirs is evaluated as $\mathcal{O}\left(\ell\lambda^2|\lambda|^2\right)$ in Tab. 2. In a similar manner, the asymptotically computational times of the joining protocol, GSig, Open and Judge are faster than those of the naive construction. Nevertheless, Open and Judge of both require the high-cost computations, namely the multiplication. It remains open whether or not a totally efficient lattice-based GSdT can be constructed.

We next discuss the efficiency of $\mathsf{GSdT}_{\mathsf{lat}}$ with [5]. For the time efficiency, all the algorithms of [5] seem faster than those of $\mathsf{GSdT}_{\mathsf{lat}}$, because they only employ the symmetric key primitives. Although the size of the signature by $\mathsf{GSdT}_{\mathsf{lat}}$ is longer than that by [5], the sizes of ok, gpk and gsk by ours are independent of the number for the attribute universe. In this sense, [5] requires one to restrict such a number to be polynomial size. Moreover, our proposed GSdT realizes not only that it has the anonymity stronger than [5] as mentioned in Section 1.1, but also in the partially dynamic model and supports the class of t-CNF as an access structure for the tracing function of openers, while [5] was merely realized in the static model and supports the class of all-and formulas.

6 Concluding Remarks

In this paper, we have introduced the first lattice-based GSdT scheme that has full anonymity. Although there exists a generic construction of GSdT from [2, 3], we take a different approach to the construction. The generic construction of [2, 3] is in the sign-then-encrypt-then-prove paradigm like the construction of the ordinary group signature [9], whereas we have employed the encrypt-then-prove paradigm, which leads to a simpler construction than the one by the generic construction. More concretely, we have employed the lattice-based CP-ABE by Tsabary [29] and the lattice-based GS by [22] (LLMNW GS) in our construction. Our result is not only the first lattice-based GSdT scheme but also gives a new technique for constructing a GSdT scheme.

We have also compared the asymptotical efficiency of our proposed scheme with the lattice-based construction, which is yielded by applying the generic construction [3] to the Tsabary CP-ABE and the pair of the signature and the NIZK which is the same as LLMNW GS, and the GSdT from symmetric-key primitives [5]. As a result, we can find that the sizes of a group public key, an opening key, a group secret key and a group signature of ours are significantly shorter than those of [3]. The computational times of joining, signing, opening and judging are asymptotically more efficient than theirs. On the other hand, comparing ours with [5], ours realizes a post-quantum construction not only which has the anonymity stronger than [9], but also in the (partially) dynamic model [9] as the original syntax by [3] for the class of richer access structures. Moreover, the sizes of keys for ours are independent of the size of the attribute universe.

We finally explain open questions. The first one is constructing lattice-based GSdT with richer access structures. Recall that access structures supported in our GSdT are only conjunctive normal

forms whose clauses have t bits of input (t-CNF). Therefore, richer access structures enable us to more flexibly designate openers during the group signing process. The second one is enhancing the efficiency of opening and judging. As in Table 2, opening and judging require many multiplications which are a high cost computation on lattices. Thus, reducing the number of multiplications on lattices is expected.

Acknowledgments

We sincerely appreciate all the anonymous reviewers for their valuable comments and suggestions on this paper and our conference version [6]. This work was supported by JSPS KAKENHI Grant Numbers JP23K11105, JP23K11106 and JP22K12023.

References

[1] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, 2011.

[2] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Group signatures with designated traceability. In *2021 Ninth International Symposium on Computing and Networking (CANDAR)*, The 9th International Symposium on Computing and Networking (CANDAR2021), pages 74–80, November 2021.

[3] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Group signatures with designated traceability over openers' attributes. *International Journal of Networking and Computing*, 12(2):493–508, July 2022.

[4] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Group signatures with designated traceability over openers' attributes in bilinear groups. In Ilsun You and Taek-Young Youn, editors, *Information Security Applications*, The 23rd World Conference on Information Security Applications (WISA 2022), pages 29–43, Cham, August 2022. Springer Nature Switzerland.

[5] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Group Signatures with Designated Traceability over Openers' Attributes from Symmetric-Key Primitives . In *2024 21st Annual International Conference on Privacy, Security and Trust (PST)*, pages 1–9, Los Alamitos, CA, USA, August 2024. IEEE Computer Society.

[6] Hiroaki Anada, Masayuki Fukumitsu, and Shingo Hasegawa. Group signatures with designated traceability over openers' attributes from lattices. In *2024 Twelfth International Symposium on Computing and Networking (CANDAR)*, pages 1–10, November 2024.

[7] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[8] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 614–629, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[9] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, pages 136–153, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[10] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 95–126, Cham, 2022. Springer International Publishing.

[11] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, and Christoph Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015.

[12] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on ddh. In Günther Pernul, Peter Y A Ryan, and Edgar Weippl, editors, *Computer Security – ESORICS 2015*, pages 243–265, Cham, 2015. Springer International Publishing.

[13] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 523–552, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[14] David Chaum and Eugène Van Heyst. Group signatures. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'91, page 257–265, Berlin, Heidelberg, 1991. Springer-Verlag.

[15] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012*, pages 60–79, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[16] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. Association for Computing Machinery.

[17] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 34–65, Cham, 2018. Springer International Publishing.

[18] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, pages 372–389, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[19] Markulf Kohlweiss and Ian Miers. Accountable metadata-hiding escrow: A group signature case study. *Proc. Priv. Enhancing Technol.*, 2015(2):206–221, 2015.

[20] Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation. In Hugo Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, pages 345–361, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[21] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 62–91, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[22] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 373–403, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[23] Benoit Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. Cryptology ePrint Archive, Paper 2016/101, 2016.

[24] Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Bifurcated signatures: Folding the accountability vs. anonymity dilemma into a single private signing scheme. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 521–552. Springer, 2021.

[25] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Accountable tracing signatures from lattices. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 556–576. Springer, 2019.

[26] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009.

[27] Amit Sahai. Simulation-sound non-interactive zero knowledge, 2000.

[28] Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. Group signatures with message-dependent opening. In Michel Abdalla and Tanja Lange, editors, *Pairing-Based Cryptography – Pairing 2012*, pages 270–294, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[29] Rotem Tsabary. Fully secure attribute-based encryption for t-cnf from lwe. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 62–85, Cham, 2019. Springer International Publishing.

[30] Shouhuai Xu and Moti Yung. Accountable ring signatures: A smart card approach. In Jean-Jacques Quisquater, Pierre Paradinas, Yves Deswarte, and Anas Abou El Kalam, editors, *Smart Card Research and Advanced Applications VI, IFIP 18th World Computer Congress, TC8/WG8.8 & TC11/WG11.2 Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS), 22-27 August 2004, Toulouse, France*, volume 153 of *IFIP*, pages 271–286. Kluwer/Springer, 2004.