Traceable Signatures using Lattices

Thakkalapally Preethi and Bharat Amberker
Department of Computer Science and Engineering, National Institute of Technology Warangal, India

Abstract: *Traceable Signatures is an extension of group signatures that allow tracing of all signatures generated by a particular group member without violating the privacy of remaining members. It also allows members to claim the ownership of previously signed messages. Till date, all the existing traceable signatures are based on number-theoretic assumptions which are insecure in the presence of quantum computers. This work presents the first traceable signature scheme in lattices, which is secure even after the existence of quantum computers. Our scheme is proved to be secure in the random oracle model based on the hardness of Short Integer Solution and Learning with Errors.*

Keywords: *Traceable Signatures, Lattices, Short Integer Solution, Learning with Errors*.

Received October 7, 2019; accepted May 5, 2020
https://doi.org/10.34028/iajit/17/6/15

1. Introduction

Group signatures, introduced by Chaum and Heyst [5], allow members to sign messages anonymously on behalf of their group. The identity of the signer is not revealed from signatures and can be verified by the group public-key. In-case of any dispute, a trusted party called group manager can trace the signature and reveal the identity of the signer. One of the applications of group signatures is cloud security [16].

If a particular group member is suspected of an illegal activity, then all the signatures generated by that member have to be detected. In group signatures, this is done by the group manager by opening all the signatures. This violates the privacy of all the group members and is inefficient (centralized opening by the group manager). To overcome these two drawbacks, Kiayias *et al*. [9] defined traceable signatures where in addition to the group manager opening the signatures individually, he can reveal the tracing trapdoor of a suspected group member to his agents and agents can detect all the signatures generated by that member without revoking the anonymity of remaining group members. This also improves scalability as agents can run in parallel compared to the traditional group signatures. Moreover in traceable signatures, a signer can provably claim the authorship of his own signatures. Since 2004, a few traceable signature schemes were proposed and these are insecure once quantum computers come into existence [17].

- *Lattice-based Cryptography*: Since the works of Regev [15] and Gentry *et al*. [7], lattice-based cryptography have been an exciting research area. It is a promising alternative to classical cryptography due to the following reasons: constructions based on lattices are secure even in the presence of quantum computers, involves simple operations and are based on worst-case hardness assumptions.

Gordon *et al*. [8] introduced the first lattice-based group signature scheme. Since then, several lattice-based group signature schemes with different features were proposed [10, 11, 12, 13, 14]. All these schemes are proved to be secure in random oracle model. Even in the random oracle model, the design of group signature schemes with different traceable mechanisms is a non-trivial problem. In particular, no lattice-based traceable signature scheme has been proposed so far.

- *Contribution*: We propose the first traceable signature scheme using lattices. Compared to the existing lattice-based group signature schemes, our scheme has following advantages:
    - User tracing: Agents on receiving the trapdoor of a particular group member from group manager can open all the signatures generated by that member preserving the anonymity of other members.
    - Group member can claim the ownership of its own previously generated signature preserving the privacy of remaining signatures generated by that member.
    - Group members can join dynamically. Compared to existing lattice-based group signature schemes which support dynamic joining [12, 14] the size of group public-key is efficient by log N factor, where N is the number of members in the group.

Our scheme satisfies the security requirements, defined by Kiayias *et al*. [9], based on the hardness of two average-case lattice problems: Short Integer Solution (SIS) and Learning with Errors (LWE).

- *Construction Overview*: To achieve dynamic joining, we adapt the joining protocol in [12] which allows the new members to sample their secret-keys and are validated by the group manager. If secret-keys are valid, the group manager issues the membership certificates. In our scheme, joining protocol is same as in [12] except during the membership certificate generation. In [12], to generate certificate, it uses the encoding function defined in [3] which consists of O (log N) matrices in group public-key (gpk). To decrease the size of gpk, our scheme uses the encoding function defined in [1], which consists of 3 matrices. Group manager maintains a database that contains all the information about registered members.

During signature generation, signer *i* generates the syndrome on its secret-key and its certificate. These syndromes are individually encrypted using Regev encryption scheme [7]. Commitment on syndrome formed by secret-key is generated using SIS function (one-way).

An interactive zero-knowledge protocol is constructed to prove signer is a valid group member, ciphertexts are well-formed and commitment generated on secret-key syndrome is the correct commitment. This protocol is repeated many times to make soundness error negligible and is made non-interactive using Fiat-Shamir heuristic [6]. Group manager possesses the secret-key of regev encryption scheme. To achieve signature opening, group manager decrypts the syndrome on secret-key and reveals the identity using the database (containing the syndromes of all secret-keys along with the identities). If all the signatures generated by a particular suspected user has to be revealed, then group manager generates the trapdoor of user *i*, syndrome on user *i* certificate and an intermediate key that decrypts the ciphertext on this syndrome, and is given to the agents. Agents upon receiving trapdoor for user *i*, decrypts the ciphertext given in the signature to obtain the syndrome on certificate *i* and matches with the syndrome given in the trapdoor. Thus, user tracing is achieved in our scheme. Signer can claim the signature as his own by generating the Non-Interactive Zero-Knowledge (NIZK) protocol that the commitment in the signature is generated by using its own secret-key. Verifiers check the validity of the protocol to verify signature claiming.

- *Organization*: In section 2, model of traceable signatures and cryptographic primitives in lattices is presented. Section 3 presents the interactive zero-knowledge protocol used in our work. Construction of our scheme and its security proofs are discussed in sections 4 and 5, respectively. Finally, section 6 concludes our work.

2. Preliminaries

2.1. Traceable Signatures

This section presents the model of traceable signature [9]. It consists of following nine algorithms.

- Setup: On input security parameter $n \in N$, a trusted party executes this algorithm and outputs the group public-key (gpk) and a group manager secret-key (*gmsk)*.
- Join: It is an interactive protocol between Group Manager (GM) and user $i(U_i)$. At the end of the protocol U_i obtains the secret-key sec_i and a membership certificate $cert_i$. GM appends the U_i transcript $transcript_i$ to the database called *transcripts*, which is a private database containing the transcripts of all users.
- Sign: On input message *m*, secret-key sec_i and membership certificate $cert_i$ this algorithm generates the traceable signature $\sum$ on *m*.
- Verify: This algorithm returns 0 or 1 when group public-key *gpk*, message *m* and signature $\sum$ are given as input.
- Open: Given a valid traceable signature $\sum$, GM using his own secret-key *gmsk* and the database *transcripts* outputs an identity of the signer.
- Reveal: Given an index *i* of a group member along with its join transcript $transcript_i$. GM using his own secret-key outputs the tracing trapdoor $trace_i$ of user *i*.
- Trace: Given a group public-key *gpk*, a valid signature $\sum$, and tracing trapdoor $trace_i$ of user *i* as input, this algorithm return 1 or 0.
- Claim: On input *gpk*, a message signature pair $(m, \sum)$ given by user *i*, user *i* secret-key sec_i and its membership certificate $cert_i$ this algorithm returns the claim τ for an authorship of *i* for signature $\sum$.
- Claim-Verify: Given a *gpk*, message-signature pair $(m, \sum)$ and claim τ, it returns 1 or 0.

- *Correctness*: A traceable signature scheme is correct if the following four conditions are satisfied with high probability in *n*, where *n* is the security parameter. Let $Sign_U$, $Reveal_U$ and $Claim_U$ be the oracles of Sign, Reveal and Claim algorithms of user *U* respectively.

a) Sign Correctness: For all *m*, Verify $(m, gpk, Sign_U)=1$.

b) Open Correctness: For any *m*, Open $(Sign_U, gpk, m, gmsk, transcripts)= U$.

c) Trace Correctness: For any *m*, Trace$(gpk, Sign_U, Reveal_U)=1$ and for any $i' \neq U$ Trace$(gpk, Sign_{i'}, Reveal_U)=0$.

d) Claim-Verify Correctness: For all $(m, \sum) \leftarrow Sign_U$ Claim—Verify $(m, \sum, Claim_U, gpk)=1$

Security model of traceable signatures was formalized in [9]. A traceable signature scheme is secure if it is secure against misidentification, anonymity and framing attacks. In all these attacks, adversary is given access to the certain oracles which share the following variables:

- *State*: contains transcripts, secret-keys and certificates of all members joined in the group. *Sigs*: set of members whose signatures are revealed by Q_{sig} query. *Revs*: set of members whose trapdoor is revealed by the Q_{reveal} query. *N* is the number of members in the group. $U^{(p)}$: set of honest members in the group. $U^{(a)}$: set of adversary controlled members in the group and $U^{(b)}$: set of members added by the adversary acting asgroup Manager (GM).

Oracles which are given access to the adversary are:

- Q_y: returns *gpk*. Q_s returns *gmsk*. $Q_{a\text{-}join}$: In the join protocol, oracle acts as a group manager and adversary acts as a user. $Q_{b\text{-}join}$: In the join protocol, adversary acts as a group manager and oracle acts as a user. When protocol in $Q_{a\text{-}join}$ and $Q_{b\text{-}join}$ terminates, it adds user *i* to $U^{(a)}$ and $U^{(b)}$ respectively and sets *state*=*state* || (*i*, $cert_i$, $transcript_i$, ⊥), *transcripts*= *transcripwts* || (*i*, $transcript_i$).
- $Q_{p\text{-}join}$: Introduces honest users in the group and sets *state* and *transcripts* as in $Q_{b\text{-}join}$ query.
- Q_{sig}: On input message *m* and index *i*, this oracle returns the signature ∑, if an entry is found in $state$ and adds (*i*, *m*, ∑) to *sigs*. If no entry is found or $i \in U^{(a)}$ then, it returns ⊥ and Q_{reveal}: returns the output of Reveal (*i*, *transcripts*) and adds *i* to *Revs*. Outputs ⊥ if $i \in U^{(b)}$ or does not exist.

- *Misidentification attack*: In this attack, adversary can control a set of users in the group through $Q_{a\text{-}join}$ query. It is allowed to observe the system while generating signatures and adding users through Q_{sig} and $Q_{b\text{-}join}$ queries.In-addition, adversary is allowed to access Q_{reveal} which reveals the tracing trapdoor of users. Finally, adversary has to generate a valid signature that is not opened or traced to a user controlled by the adversary. It can be clearly explained in the following experiment.

Experiment $Exp_A^{mis}(n)$:$(gpk, gmsk) \leftarrow Setup(1^n)$; $(m, \Sigma) \leftarrow A\big(Q_{p-join}, Q_{a-join}, Q_{reveal}, Q_{sig}\big)$; If Verify$(m, \Sigma, gpk) = 0$ then return 0; If Open$(m, \Sigma, gmsk) =$ j $\notin U^{(a))}$ or $\wedge_{i\in U^{(a)}}$ Trace$\big(\Sigma,$ Reveal(i)$\big) = 0$ then return 1; return 0; A traceable signature is secure against misidentification attacks if $\Pr\big[Exp_A^{mis}(n) = 1\big]$is negligible in *n*.

- *Anonymity attack*: This attack operates in two phases: play and guess. In play phase, adversary has access to $Q_{a-join}, Q_{p-join}, Q_{sig}\ and\ Q_{reveal}$ through which it controls set of users, observes the system during addition of members and signature generation and can obtain the tracing information of any user. At the end of play phase, adversary chooses two honest users which are not input to Q_{reveal} query and obtains signature generated by one of them. In the guess stage, adversary has to guess the identity of the signer. This can be explained with the following experiment.

Experiment $Exp_A^{anon}(n)$: $(gpk, gmsk) \leftarrow Setup(1^n)$; $(aux, m, i_0, i_1) \leftarrow A\big(Q_{p-join}, Q_{a-join}, Q_{reveal}, Q_{sig}\big)$; If $i_0 \notin U^{(p)}$ or$i_1 \notin U^{(p)}$ or $i_0 \in Revs$or$i_1 \in Revs$then return 0; $b \leftarrow \{0,1\}, \Sigma \leftarrow$ Sign$(gpk, m, sec_{i_b}, cert_{i_b})$; b'←A$(guess, aux, \Sigma\text{:}\ Q_{p-join}, Q_{a-join}, Q_{reveal}, Q_{sig})$I f b=$b'$, then return 1; return 0; A traceable signature is said to be secure against anonymity attacks if for any probabilistic polynomial-time algorithm *A*, $|\Pr[Exp_A^{anon}(n) = 1] - \frac{1}{2}\ |$ is negligible in n.

- *Framing attacks*: In this attack, adversary is allowed to control group manager through Q_S query. It can observe the system through $Q_{b\text{-}join}$ and Q_{sig} queries. The goal of the adversary is either to generate a signature that opens or traces to honest user or to claim the ownership of the signature generated by another user. It can be described by the following experiment.

Experiment $Exp_A^{fra}(n)$: $(gpk, gmsk) \leftarrow Setup(1^n)$; $(m, \Sigma, \tau) \leftarrow$ A$(Q_y, Q_S, Q_{b-join}, Q_{sig})$; If Verify$(m, \Sigma, gpk) = 0$ then return 0; If Open$(m, \Sigma, gmsk) \in U^{(b)}$ or $\forall_{i\in U^{(b)}}$Trace$\big(\Sigma,$ Reveal(i)$\big) = 1$then return 1; If $\forall_{i\in U^{(b)}}(i, \Sigma) \in sigs$ and Claim Verify$(m, \Sigma, \tau, gpk) = 1$ then return 1; return 0; A traceable signature is secure against framing attacks if for any probabilistic polynomial-time adversary *A*, $\Pr\big[Exp_A^{fra}(n) = 1\big]$ is negligible in *n*.

2.2. Lattices

For any *m* linearly independent vectors B= (b_1, ..., b_m**)**, lattice L(*B*) is defined as

$$L(B) = \{\sum_{i=1}^{m} x_i\, b_i\text{:}\, x_i\ \in Z\}.$$

For any positive real number s, discrete guassian distribution over lattice Λ is defined as $D_{\Lambda,s}(x) = \rho_s(x)/\rho_s(\Lambda)$for any $x \in \Lambda$.

For any $m, n\ \geq 1, q \geq 2$, matrix $A\ \in Z_q^{n\times m}$, lattice $\Lambda^{\perp}(A)$ is defined as

$$\Lambda^{\perp}(A) = \{e\ \in Z^m\text{:}\, Ae = 0\ mod\ q\}$$

For any $u \in Z_q^n$, coset of the lattice $\Lambda_u^{\perp}(A)$ is defined as

$$\Lambda_{\boldsymbol{u}}^{\perp}(A) = \{\boldsymbol{e} \in Z^m : A\boldsymbol{e} = \boldsymbol{u} \bmod q\}$$

In our work, we consider two average case lattice problems are Short Integer Solution (SIS) and Learning With Errors (LWE).

$\mathrm{SIS}^{p}_{n,m,q,\beta}$: Given a uniformly random matrix $A \in Z_q^{n\times m}$, find the vector $x \in \Lambda^{\perp}(A)$ such that $||x||_p \leq \beta$.

$\mathrm{LWE}_{n,q,\psi}$: Let $n,m \geq 1, q \geq 2$ and ψ be the probability distribution over Z. For $\in Z_q^n$, the distribution $A_{s,\psi}$ over $Z_q^n \times Z_q$ is obtained by sampling a uniform vector $a \in Z_q^n$, $e \in \psi$ and outputting the pair $(a, a^T s + e)$. The goal of $\mathrm{LWE}_{n,q,\psi}$ is to distinguish m samples chosen according to $A_{s,\psi}$ from the m samples chosen according to uniform distribution over $Z_q^n \times Z_q$.

3. Underlying Zero-Knowledge Argument System

Let *D*, *L* be positive integers. Libert *et al*. [12] proposed an interactive zero-knowledge protocol for the relation *R*

$$R = \{(P, y; x) \in Z_q^{D\times L} \times Z_q^D \times Valid : Px = y \bmod q\} \quad (1)$$

Where $Valid$ is the subset of $\{-1,0,1\}^L$ satisfying the following two conditions:

1) $\boldsymbol{x} \in Valid \Leftrightarrow T_\pi(\boldsymbol{x}) \in Valid$

2) If $\boldsymbol{x} \in Valid$ and π is uniform in S then $T_{\pi(x)}$ is uniform in $Valid$

Where T_π is the permutation of *L* elements and set *S* is the permutation of *m* elements.

This section presents the Zero-knowledge Argument of knowledge (ZKAoK) for the scheme in section 4. In detail, it presents ZKAoK that satisfies the following conditions:

- Signer *i* is a certified group member i.e, he possess a valid secret-key z_i and membership certificate $cert_i=(i,d_i,s_i)$
- The syndrome v_i obtained using secret-key z_i is correctly encrypted to ciphertext $c_{v_i}=(c_1,c_2)$.
- The syndrome w_i obtained using $cert_i$ and z_i is correctly encrypted to ciphertext $c_{w_i}=(c_3,c_4)$..
- Commitment b_1 is the correct commitment of v_i.

All the above conditions can be defined as a relation R'.

- *Definition* 1: The relation R' is defined as

$$R' = \{A, A_1, A_2, u, F, B, B_1, D, D_0, G_0, G_3, D_1, c_1, c_2, c_3, c_4, b_1; i, z_i, v_i, w_i, d_i, s_i, s_0', s_1', x_1, x_2, x_3, x_4\}$$

Where

$$A, A_1, A_2, B, F, D, G_3 \in Z_q^{n\times m}, D_0, D_1 \in Z_q^{2n\times 2m}, F \in Z_q^{4n\times 4m}, B_1, G_0 \in Z_q^{n\times 2m}, u \in Z_q^n, c_1, c_3, c_4 \in Z_q^m, c_2 \in Z_q^{2m}; i \in [N], z_i, d_i, s_i \in [-\beta,\beta]^{2m}, v_i \in Z_q^{4n}, w_i \in Z_q^{2n}, s_0', s_1' \in [-b,b]^n, x_1, x_3, x_4 \in [-b,b]^m, x_2 \in [-b,b]^{2m} \quad (2)$$

Satisfying

$$Ad_{1i} + A_1 d_{2i} + iA_2 d_{2i} = u + Dbin(w_i)$$
$$w_i = D_0 bin(v_i) + D_1 s_i \bmod q \text{ and } v_i = Fz_i \bmod q \quad (3)$$

$$c_{v_i} = (c_1, c_2) = \left(B^T s_0' + x_1, G_0^T s_0' + x_2 + bin(v_i)\frac{q}{2}\right)$$
$$c_{w_i} = (c_3, c_4) = \left(B^T s_1' + x_3, G_3^T s_1' + x_4 + bin(w_i)\frac{q}{2}\right) \quad (4)$$

$$b_1 = B_1 bin(v_i) \bmod q$$

Since, Libert *et al*. [12] proposed an interactive zero-knowledge protocol for relation R, an interactive zero-knowledge protocol for relation R' can be generated by transforming the relation R' to relation R (defined in Equation (1)).

3.1. Transformation of R′ to R

To transform the relation to *R*, we transform Equations (2), (3), and (4) to the form $Px = y \bmod q$, and define a set $Valid$ such that it satisfies the conditions (1) and (2). We define the sets and matrices which are used in the transformation.

- B_{3m} is the set of all vectors in $\{-1,0,1\}^{3m}$ having equal number of -1,0,1. B_{2l} is the set of all vectors in $\{0,1\}^{2l}$ having hamming weight l.
- For any $\alpha > 0$, one can define the sequence $(\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_p)$ such that $\sum_{i=1}^{p} \alpha_i = \alpha$ where $p = \log\beta + 1$ [11]. A matrix $H_{m,\alpha}$ is defined as $[\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_p] \otimes I_m \in Z^{m\times mp}$ and a matrix $H_{m,\alpha}^*$ is obtained by adding $2m$ columns to $H_{m,a}$.
- We define the matrix R_1 as $R_1 = I_{4n} \otimes [1|2|4| \ldots |2^{\log q} - 1]$ and R_2 as $R_2 = I_{2n} \otimes [1|2|4| \ldots |2^{\log q} - 1]$.

The following lemma is used in the transformation.

- *Lemma* 4 [13]: Let *m,O* be positive integers and $\delta_O = \log O + 1$. On input a vector $v \in [-0, 0]^m$, extension and decomposition technique outputs a vector $v^* \in B_{3m\delta_O}$ such that $H_{m,O}^* v^* = v$

Conversion of all the equations in definition 1 to $Px = y \bmod q$ proceeds as follows:

- Transformation of Equation (2) to the appropriate form: Let $id \in \{0,1\}^l$ is the binary representation of *i* and id_j represents the *j*-th bit of *id*. Let $y_1=bin(v_i)$ and $y_2= bin(w_i)$ and Equation (2) can be written as

$$Ad_{1i} + A_1 d_{2i} + \sum_{i=1}^{l}(2^{l-i}A_2) id_i d_{2i} - Dy_2 = u \quad (5)$$

$$D_0 y_1 + D_1 s_i - R_2 y_2 \bmod q = 0, \; R_1 y_1 - Fz_i = 0 \bmod q \quad (6)$$

Apply lemma (4) to the vectors $d_{1i}$and d_{2i} to generate the vectors d_{1i}^* and d_{2i}^* respectively. Extend$y_2 \in \{0,1\}^m$ and $id \in \{0,1\}^l$to $\widehat{y_2}$ andid^*such that $\widehat{y_2} \in B_{2m}$ and $id^* \in B_{2l}$. Now, Equation (5) is reduced to

$$A^* x_{11} = u \bmod q \quad (7)$$

Where

$$A^* = [AH^*_{m,\beta}|A_1 H^*_{m,\beta}|2^{l-1}A_2 H^*_{m,\beta}|\ldots|2^0 A_2 H^*_{m,\beta}|-D|0^{n\times m}]$$

And

$$x_{11} = [d_{1i}^*||d_{2i}^*||id_1^* d_{2i}^*||\ldots||id_{2l}^* d_{2i}^*||\hat{y}_2]$$

Similarly, Equation (6) is reduced to

$$Cx_{12} = 0 \bmod q \quad (8)$$

Where

$$C = \begin{pmatrix} C_1 & 0 \\ 0 & C_2 \end{pmatrix} \text{ and } x_{12} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

And $C_1 = [D_0|0^{2n\times 2m}|D_1 H^*_{m,\beta}|-R_2|0^{n\times m}]C_2 = [R_1|0^{2n\times 2m}|-FH^*_{m,\beta}]$,$t_1 = [\widehat{y_1}||s_i^*||\widehat{y_2}]$ and $t_2 = [\widehat{y_1}||z_i^*]$. The vectors $s_i^*$and z_i^* are obtained by applying lemma (4) to s_i and z_i respectively and $\widehat{y_1}$ is obtained by extending y_1 such that $\widehat{y_1} \in B_{4m}$.

We combine Equations (7), (8) to obtain

$$P_1^* x_1^* = z_1 \bmod q \quad (9)$$

Where

$$P_1^* = \begin{pmatrix} A^* & 0 \\ 0 & C \end{pmatrix} x_1^* = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} z_1 = \begin{pmatrix} u \\ 0 \end{pmatrix}$$

- Transformation of Equation (3) to the required form: Equation (3) can be written as

$$\begin{pmatrix} 0 \\ \frac{q}{2}\boldsymbol{I}_{2m} \\ 0 \\ 0 \end{pmatrix} y_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{q}{2}\boldsymbol{I}_m \end{pmatrix} y_2 + \begin{pmatrix} \begin{matrix} B^T \\ G_0^T \end{matrix} \Big| I_{3m} \Big| 0 \\ 0 \Big| \begin{matrix} B^T \\ G_3^T \end{matrix} \Big| I_{2m} \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_0' \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{s}_1' \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{pmatrix}$$

$$\equiv Q_1 y_1 + Q_2 y_2 + Q_3 t_3 = z_2 \quad (10)$$

Apply lemma (4) to the vector t_3 to generate $t_3^* \in B_{3(2n+5m)\delta_b}$ and $\widehat{y_1} \in B_{4m}$, $\widehat{y_2} \in B_{2m}$ is obtained by extending y_1 and y_2 respectively. Equation (10) can be written as

$$P_2^* x_2^* = z_2 \quad (11)$$

Where

$$P_2^* = [Q_1|0^{5m\times 2m}|Q_2|0^{5m\times m}|Q_3 H^*_{m,b}],$$
$$\boldsymbol{x}_2^* = [\widehat{\boldsymbol{y}_2}||\widehat{\boldsymbol{y}_1}||\boldsymbol{t}_3^*]$$

- Transformation of Equation (4) to the required form: Let $B_1^* = [B_1|0^{n\times 2m}]$ and $\widehat{y_1} \in B_{4m}$ is obtained by extending $y_1 \in \{0,1\}^{2m}$. Therefore, Equation (4) can be written as

$$P_3^* x_3^* = z_3 \quad (12)$$

Where $P_3^* = B_1^*, x_3^* = \widehat{y_1}$ and $z_3 = b_1$.

Finally, we combine the Equations (9), (11), and (12) as follows: Generate the matrix P, x and y as

$$\mathrm{P} = \begin{pmatrix} P_1^* & 0 & 0 \\ 0 & P_2^* & 0 \\ 0 & 0 & P_3^* \end{pmatrix} x = \begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \end{pmatrix} \text{ and } \mathrm{y} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \quad (13)$$

Thus, all the equations in relation R' (definition 1) are transformed to the form $Px = y \bmod q$.

Let $L = 22m + (2l+4)3m\delta_\beta + 3(2n+5m)\delta_b$.

We define set *Valid* as follows:

$Valid$: Set of all vectors $\{-1,0,1\}^L$ of the form

$g = [g_1||g_2||t_1 g_2||\ldots||t_{2l} g_2||g_3||g_4||g_5||g_3||g_4||g_6$
$g_3||g_4||g_7||g_4]$

Where $g_1, g_2, g_5, g_6 \in B_{3m\delta_\beta}, g_3 \in B_{2m}, g_4 \in B_{4m}, g_7 \in B_{3(2n+5m)\delta_b}, t \in B_{2l}$.

Let $S = S_{3m\delta_\beta} \times S_{3m\delta_\beta} \times S_{2l} \times S_{2m} \times S_{4m} \times S_{3m\delta_\beta} \times S_{3m\delta_\beta} \times S_{3(2n+5m)\delta_b}$

Let $\pi = (\pi_1, \pi_2, \tau, \pi_3, \pi_4, \pi_5, \pi_6, \pi_7) \in S$.Define the permutation T_π as

$T_\pi(g)=[\pi_1(g_1)\|\pi_2(g_2)\|t_{\tau(1)}(\pi_2(g_2))\|\ldots\|$
$t_{\tau(2l)}(\pi_2(g_2))||\pi_3(g_3)\|\pi_4(g_4)\|\pi_5(g_5)\|\pi_3(g_3)\|$
$\pi_4(g_4)||\pi_6(g_6)\|\pi_3(g_3)\|\pi_4(g_4)||\pi_7(g_7)\|\pi_4(g_4)]$

By construction of vector *x* in section 3.1, it belongs to set *valid*. It can be observed if a vector $x \in Valid$then $T_{\pi(x)} \in Valid$ and vice-versa. Therefore, both the conditions (1 and 2) for *valid* set are satisfied. Since ZKAoK protocol for relation *R* is given in [12] and our relation *R´* is transformed to *R*, ZKAoK protocol for relation *R´* is directly constructed from *R*.

4. Proposed Scheme

For any two matrices *A* and *B*, concatenation of rows and columns are represented by [*A*|*B*] and [*A*||*B*] respectively. Similar notation is also used for vectors.

We assume each user U_i has public-key *upk*[*i*] and secret-key of *upk*[*i*] of a signature scheme as in [9]. Let *n* be the security parameter, *N* is the number of users joined the group, $m = 2n\log q, q = \tilde{O}(ln^3)$ and $q \gg N, \sigma = \Omega(\sqrt{n\log q}\log n), \beta = \sigma\omega(\log m)$. Let $b = \sqrt{n}\omega(\log n), t = \omega(\log n)$and ψ be the *b*-bounded distribution. We consider three random oracles $H: \{0,1\}^* \to \{0,1,2\}^t$, $H_1: \{0,1\}^* \to Z_q^{n\times m}$ and $H_2: \{0,1\}^* \to Z_q^{n\times 2m}$. We use GenTrap and SamplePre algorithms presented in [2, 7] for our construction.

- *Setup* (1^n)
  1. Generate two instances of hard random lattices (A,T_A) and (B,T_B) using algorithm GenTrap (n,m,q).
  2. Choose matrices (A_1,A_2,D) uniformly over $Z_q^{n\times m}$, *F* is sampled uniformly from $Z_q^{4n\times 4m}$, (D_0,D_1) is uniformly chosen over $Z_q^{2n\times 2m}$, B_1 is uniformly chosen over $Z_q^{n\times 2m}$ and vector u is chosen uniformly over Z_q^n.

$gpk = (A, A_1, A_2, B, B_1, D, D_0, D_1, F, u)$ and $gmsk = (T_A, T_B)$

Note: The size of gpk is $O(nm\log q)$.

Join (*GM*,U_i).

1. User U_i chooses a vector $z_i \leftarrow D_{Z^{4m},\sigma}$ and compute a vector $v_i=Fz_i mod\ q$. Generate a signature on v_i i.e., $sig_i = \text{sign}_{usk[i]}(v_i)$. Send v_i and sig_i to GM.
2. GM verifies sig_i is valid signature of vector v_i using $upk[i]$ and was not previously generated by another user. If it is valid, then GM sets $i=N+1$ and computes user dependent matrix A_i as $A_i = [A|A_1 + iA_2] \in Z_q^{n\times 2m}$ and short vector $d_i = [d_{1i}||d_{2i}] \in Z^{2m}$ such that

$$A_i d_i = u + u_i \ mod\ q \tag{14}$$

Where $u_i = D\ bin(D_0 bin(v_i) + D_1 s_i)$ and s_i is chosen according to $D_{Z^{2m},\sigma}$ and send (i, d_i, s_i) to U_i

3. U_i checks whether (i, d_i, s_i) satisfies Equation (14), $||d_{ji}||_\infty \leq \beta$ for $j \in \{1,2\}$ and $||s_i||_\infty \leq \beta$

If the conditions are valid then, $sec_i = z_i, cert_i = (i, d_i, s_i)$ and stores the $transcript_i = (sig_i, v_i, i, d_i, s_i, upk[i])$ in the database *transcripts* which is the private database of GM.

Sign $(gpk, cert_i, sec_i, m)$

1. Generate the one-time signature key-pair (*VK*, *SK*) Compute $v_i=Fz_i\ mod\ q$ and $w_i=D_0 bin(v_i)+D_1 s_i\ mod\ q$.
2. Encrypt the vector v_i using dual Regev Encryption scheme [7]. Let $G_0 = \text{H}_2(VK)$. Choose $s_0' \leftarrow \psi^n$, $x_1 \leftarrow \psi^m$ and $x_2 \leftarrow \psi^{2m}$.

$$c_{v_i} = (c_1, c_2) = (B^T s_0' + x_1, G_0^T s_0' + x_2 + bin(v_i)\left(\tfrac{q}{2}\right)) \tag{15}$$

3. Similarly, encrypt the vector w_i Let $G_1=\text{H}_1(cert_i)$. Choose $s_1' \leftarrow \psi^n$ and compute $G_2 \in Z_q^{m\times n}$ such that $G_2 s_1'=0$ mod q and proceed if one such G_2 is found otherwise repeat. Let $G_3=G_1+G_2^T$. Choose $x_3, x_4 \leftarrow \psi^m$

$$c_{w_i} = (c_3, c_4) = \left(B^T s_1' + x_3, G_1^T s_1' + x_4 + bin(w_i)\left(\tfrac{q}{2}\right)\right) \tag{16}$$

4. Generate the commitment for $\boldsymbol{v}_i$ as

$$b_1 = B_1 bin(v_i)\ mod\ q \tag{17}$$

5. Generate a NIZK protocol Π to prove there exists $i \in [N]$, $(z_i, d_{1i}, d_{2i}, s_i)$ has infinity bound β, $(s_0', s_1', x_1, x_2, x_3, x_4)$ has infinity bound b and there exists v_i and w_i that satisfies Equations (2), (3) and (4). This can be generated by running the interactive protocol in section 3 t times and converting it into non-interactive using Fiat-Shamir heuristic [6] i.e., Π=(CMT,CH,RSP} where $\text{CH}(\text{ch}_1 \ldots, \text{ch}_t)\text{H}(\text{CMT}, \text{m}, \{c_i\}_{i=1}^4, \text{VK}, G_3, b_1) \in \{0,1,2\}^t$.
6. Compute the one-time signature $sig = \text{OSign}(\text{SK}, (\{c_i\}_{i=1}^4, b_1, \Pi))$

$$\Sigma = (c_{v_i}, c_{w_i}, \Pi, VK, sig, G_3, b_1) \tag{18}$$

Verify(m, gpk, ∑).

1. Check whether protocol Π is valid.
2. Check whether *sig* is a valid signature on $(\{c_i\}_{i=1}^4, b_1, \Pi)$ using *VK*.

Return 1 iff all the conditions are valid.

Open (∑, gpk, m, gmsk, transcripts)

1. Compute $G_0=\text{H}_2(VK)$ Decrypt c_{v_i} using T_B as follows: Using T_B, compute a small-norm matrix $E_0 \in Z^{m\times 2m}$ such that $BE_0=G_0\ mod\ q$. Obtain $bin(v_i)$ by computing $\left(c_2 - \frac{E_0^T c_1}{\frac{q}{2}}\right)$
2. Compute $v_i = R_1 \otimes bin(v_i)$ and search in the database *transcripts* for $transcript_i$ in which v_i is the entry. If such *transcript* is found output the signer i otherwise output ⊥.

Reveal (gmsk, i, transcripts)

1. Parse $gmsk=(T_A, T_B)$ and obtain $transcript_i = (sig_i, v_i, i, d_i, s_i, upk[i])$ from the database $transcripts$. Compute $w_i = D_0 bin(v_i) + D_1 s_i mod\ q$
2. Let $G_1=\text{H}_1(i,d_i,s_i)$. Compute the small norm matrix $E_i \in Z^{m\times m}$ using T_B such that $BE_i=G_1\ mod\ q$.

   $trace_i=(w_i,E_i)$.

Trace (gpk,∑, $trace_i$)

1. Parse $trace_i=(w_i,E_i)$ and signature $\Sigma = (c_{v_j}, c_{w_j}, \Pi, VK, sig, G_3, b_1)$.
2. Decrypt c_{w_j} using E_i and obtain $bin(w_j)$.
3. Compute $w_j = R_2 \otimes bin(w_j)$ and return 1 iff w_j is equal to the w_i which is given as a part of $trace_i$.

Claim (m,∑,sec_i,$cert_i$,gpk)

1. Parse the signature $\Sigma = (c_{v_i}, c_{w_i}, \Pi, VK, sig, G_3, b_1)$
2. Generate the NIZK proof of knowledge π that there exists z_i such that

$$b_1 = B_1 bin(v_i)\ mod\ q \text{ where } v_i = Fz_i\ mod\ q$$

This is possible only if user has the secret-key z_i and generated the signature Σ.
Output: $\tau = \pi$

Claim Verify(m, Σ, τ, gpk)

Parse $\tau = \pi$ and signature $\Sigma(c_{v_i}, c_{w_i}, \Pi, VK, sig, G_3, b_1)$. Check the validity of protocol π and return 1 if it is valid.

- Correctness:
  - Sign Correctness: By completeness of protocol ∏ and correctness of one-time signature scheme, Verify algorithm returns 1 with high probability.

- Open Correctness: By correctness of Dual-Regev Encryption Scheme [7], open algorithm returns the identity U with high probability.
- Trace Correctness: We know $Sign_U$ and $Reveal_U$ oracles generate the signature and tracing trapdoor of user U respectively. By Reveal algorithm in section 4, the tracing trapdoor is (w_U,E_U). By correctness of dual Regev encryption scheme, $E_U$returnsw_U. Therefore, Trace (gpk,$Sign_U$,$Reveal_U$)=1 is satisfied with high probability. Next, we need to prove Trace (gpk,$Sign_{i'}$,$Reveal_U$)=0 for any $i' \neq U$. During trace algorithm in section (4), we decrypt (c_3,c_4) using E_U and obtain bin (w_i). But, (c_3,c_4) is the encryption of $w_{i'}$ and E_U which is the trapdoor of user U does not decrypt to $w_{i'}$ correctly. Assume decryption algorithm returns $w_{j'}$. Since, w_U is statistically close to uniform, the probability that $w_{j'}=w_U$ is negligible. Therefore, Trace(gpk,$Sign_{i'}$,$Reveal_U$)=1is negligible for any $i' \neq U$.
- Claim-Verify Correctness: By completeness of protocol π generated by $claim_U$, Claim-Verify algorithm returns 1 for all (m,$\sum$)←$Sign_U$.

5. Security

5.1. Misidentification Attacks

- *Theorem* 2: Our scheme is secure against misidentification attacks based on the hardness of SIS assumption.
- *proof:* Assume, there exists an adversary A breaking the security of our scheme against misidentification attacks with non-negligible probability. We construct an algorithm B that solves SIS instance $\overline{A} = [\overline{A}_1|\overline{A}_2] \in Z_q^{n\times 2m}$with non-negligible probability. A $coin$ is uniformly chosen over {1,2} and $i^* \overset{\$}{\leftarrow} [N]$.

coin=1

- *Setup:*
  - Assign the matrix $A = \overline{A}_1$. Run Gen Trap (n,m,q) and obtain (A_2, T_{A_2}) and (B,T_B). Sample the matrices R, R' uniformly over $\{-1,1\}^{m\times m}$ and compute $A_1 = AR - i^*A_2$.
  - Sample the matrices D_0, D_1 uniformly over $Z_q^{2n\times 2m}$, $D = \overline{A}_2R'$, matrix B_1 over $Z_q^{n\times 2m}$ and matrix $F \in Z_q^{4n\times 4m}$.
  - Vector e is chosen according to $D_{Z^m,\sigma}$ and compute $u = Ae \bmod q$.

Send $gpk = (A, A_1, A_2, B, B_1, D, D_0, D_1, u, F)$ to A

- *Queries*:
  - $Q_{P\text{-}join}$: Increments N and compute $A_N = [A|A_1 + NA_2]$. Using T_{A_2} obtain $d_N = [d_{N,1}||d_{N,2}]$. Let s_N is chosen according to $D_{Z^{2m},\sigma}$ and z_N are chosen according to $D_{Z^{4m},\sigma}$. Let $cert_N$=(N,d_N,s_N), sec_N=z_N and add N to the set $U^{(p)}$.
  - $Q_{a\text{-}join}$: When A triggers join protocol by sending v_i,B chooses N such that $N \neq i^*$.When A provides sig_i such that it is a valid signature on v_i under $upk[i]$. Using T_{A_2} obtain the vector $d_N = [d_{N,1}||d_{N,2}]$ such that $A_Nd_N = u + D\big(bin(D_0\, bin(v_N) + D_1s_N)\big) \bmod q$ where s_N is chosen according to $D_{Z^{2m},\sigma}$. Send $cert_N$=(N,d_N, s_N) to A and add N to the set $U^{(a)}$.
  - Q_{sig}: If $i \notin U^{(p)}$ or $i = i^*$ then abort. Otherwise, generate the signature $\sum$ on message m using sec_i.
  - Q_{reveal}: On input index i, if $i \notin U^{(p)}$ or i=i^* then abort. Otherwise, algorithm B searches in the database $transcripts$ for the entry $(.,.,i,d_i,s_i,.)$. Using $transcripts_i$ obtain $trace_i$ and add i to $Revs$.
  - Forgery: A outputs (m^*,Σ^*)such that Verify $(\mathrm{gpk},\mathrm{m}^*,\Sigma^*)=1$. If$\mathrm{Open}((\mathrm{m}^*,\Sigma^*,\mathrm{gmsk}) = \mathrm{j} \in \mathrm{U}^{(a)}or\mathrm{j} \neq \mathrm{i}^*$abort. Otherwise, parse $\Pi^* = (\mathrm{CMT},\mathrm{CH},\mathrm{RSP})$ and $\Sigma^* = (\mathrm{c_{v_j}}^*, \mathrm{c_{w_j}}^*, \Pi^*, \mathrm{VK}^*, \mathrm{G_3}^*, \mathrm{b_1}^*, \mathrm{sig}^*)$. A must have queried the random oracle H on input $(\mathrm{CMT},\mathrm{m}^*,\mathrm{VK}^*,\{\mathrm{c}_i^*\}_{i=1}^4,\mathrm{G}_3^*,\mathrm{b}_1^*)$ with high probability. Otherwise,

$$\Pr[\{ch\}_{i=1}^t = \mathrm{H}(\mathrm{CMT}, m^*, VK^*, \{c_i^*\}_{i=1}^4, G_3^*, b_1^*)] \leq \frac{1}{3^t} \quad (19)$$

Therefore with $\varepsilon - 3^{-t}$probability, there exists an index $\kappa^* \leq Q_H$. At this stage, algorithm B runs A with same input and random tape as in original execution. Pick κ^* as the target point and replay A many times with the same random tape and input. Each time, first $\kappa^* - 1$ queries are answered as $r_1, \ldots, r_{\kappa^*-1}$ and fromκ^*th query the answers are uniformly chosen from $\{1, 2, 3\}^t$. The Improved Forking Lemma [4] implies that, with probability greater than $\frac{1}{2}$, B can obtain a 3-fork involving tuple $(\mathrm{CMT}, m^*, VK^*, \{c_i^*\}_{i=1}^4, G_3^*, b_1^*)$ and open to the $bin(v_i^*)$ which is uniquely determined by (c_1^*, c_2^*). Let the answers of B with respect to the 3-fork branches be

$$r_{\kappa^*}^{(1)} = \left(ch_1^{(1)}, \ldots, ch_t^{(1)}\right) r_{\kappa^*}^{(2)} = \left(ch_1^{(2)}, \ldots, ch_t^{(2)}\right)$$

$$r_{\kappa^*}^{(3)} = (ch_1^{(3)}, \ldots, ch_t^{(3)}) \quad (20)$$

$$\Pr\left[\exists j \in [t]: \left\{ch_j^{(1)}, ch_j^{(2)}, ch_j^{(3)}\right\} = \{1,2,3\}\right] = (1 - (\frac{7}{9})^t)$$

If such j exists, parse the 3-forgeries corresponding to 3-fork branches to obtain $(\mathrm{RSP}_j^{(1)}, \mathrm{RSP}_j^{(2)}, \mathrm{RSP}_j^{(3)})$. Given three different challenges and three valid responses for same commitment CMT_j, using witness extractionprocedure, the witness $(j, d_j =$

$[d_{1j}||d_{2j}], z_j, s_j)$ can be extracted. Algorithm B aborts if $j \neq i^*$. We know $A_j d_j = u + Dbin(w_j) \bmod q$ where $w_j = D_0 bin(v_j) + D_1 s_j \bmod q$. Therefore,

$$[\overline{A_1}|\overline{A_1}R][d_{1j}||d_{2j}] = u + Dbin(w_j) \bmod q$$
$$\overline{A_1}\,(d_{1j} + Rd_{2j} - e) - \overline{A}_2 R' bin(w_j)) = 0 \bmod q \quad (21)$$

Let $\overline{x} = [d_{1j} + Rd_{2j} - e|| - R' bin(w_j)]$. Since the vector u statistically hides e in $\Lambda_u^{\perp}(\overline{A}_1)$, $\overline{x} \neq 0$. Therefore, $\overline{x}$ is a solution to SIS instance i.e., $\overline{A}\overline{x} = 0 \bmod q$ and $||\overline{x}|| \leq \sqrt{m}(\beta(m+2)+m)$.

Coin=2

- *Setup*:
  - Assign the matrix $A = \overline{A}_1$ and $D = \overline{A}_2$. Run GenTrap (n,m,q) obtain (A_2, T_{A_2}) and (B, T_B). Compute $A_1 = AR - i^* A_2$ where R uniformly over $\{-1,1\}^{m \times m}$
  - Sample the matrix D_0 uniformly over $Z_q^{2n \times 2m}$ and matrices F, B_1 are uniformly chosen over $Z_q^{4n \times 4m} \times Z_q^{n \times 2m}$ respectively. Let (D_1, T_{D_1}) is obtained using GenTrap $(2n, 2m, q)$. Let $A_{i^*} = [A|A_1 + i^* A_2]$.
  - Let d_{1i^*} and d_{2i^*} are chosen according to $D_{Z^m,\sigma}$. Compute $u = A_{i^*} d_{i^*} - Dbin(c')$ where c' is uniformly chosen over Z_q^n.
  - Send $gpk = (A, A_1, A_2, B, B_1, D, D_0, D_1, u, F)$ to A.
- *Queries*: $Q_{P\text{-}join}$, Q_{sig} *and* Q_{reveal}: Answer similarly as in *coin*=1. For $Q_{a\text{-}join}$ query proceed as follows If $i \neq i^*$ then, proceed as in case of coin=1. If $i = i^*$: Recall d_{i^*} and c'. If A provides valid signature on v_{i^*} then using T_{D_1} obtain s_{i^*} such that $D_1 s_{i^*} = c' - D_0\, bin(v_{i^*})$. Send $cert_{i^*} = (i^*, d_{i^*}, s_{i^*})$ to A.
- *Forgery*: A outputs (m^*, Σ^*) and abort if $\text{Open}((m^*, \Sigma^*, gmsk) = \text{j} \notin U^{(a)}$ or $j \neq i^*$. Proceed if Verify (gpk, m^*, Σ^*)=1, $\text{Open}((m^*, \Sigma^*, gmsk) = \text{j} \in U^{(a)}$ and $\wedge_{i \in U^{(a)}}$ Trace $(\Sigma^*, \text{Reveal}(i))$=0. Using forking lemma and knowledge extractor we obtain (d^*, z^*, s_{i^*}). Since Trace $(\Sigma^*, \text{Reveal}(i^*))$=0, $w^* = D_0 bin(v^*) + D_1 s^* \neq D_0 bin(v_{i^*}) + D_1 s_{i^*} = w_{i^*}$ and we know $[\overline{A_1}|\overline{A_1}R][d_1^*||d_2^*] - Dbin(w^*) = [\overline{A_1}|\overline{A_1}R][d_{1i^*}||d_{2i^*}] - Dbin(w_{i^*}) = u$. Therefore, $\bar{x} = [d_1^* - d_{1i^*}||R(d_2^* - d_{2i^*})||w^* - w_{i^*}]$ is a solution to SIS instance i.e., $\overline{A}\,\bar{x} = 0 \bmod q$, $\overline{x} \neq 0$, and $||\bar{x}|| \leq \sqrt{m}(2\beta + 2m\beta + 1)$.

5.2. Anonymity Attacks

- *Theorem3:* Our scheme is secure against anonymity attacks based on the zero-knowledge property of NIZK protocol Π and hardness of LWE.
- *Proof:* To prove our scheme is secure against anonymity attacks, we define two games $G_0^{(b)}$ and G_7. Game $G_0^{(b)}$ is the original anonymity game where challenge signature is generated by one of the users and game G_7 is the anonymity game where challenge signature is generated independent of both the users. We show that challenge signatures generated in both these games are computationally indistinguishable. This is because, if signatures generated in both these games are indistinguishable, then the advantage of adversary guessing the signer is negligible. To prove the signatures generated in these two games are indistinguishable, we define intermediate games $G_1^{(b)}, G_2^{(b)}, G_3^{(b)}, G_4^{(b)}, G_5^{(b)}$ and $G_6^{(b)}$.
- *Game* $G_0^{(b)}$: This is the original anonymity game. In precise, challenger runs setup algorithm to generate (*gpk,gmsk)* and gives *gpk* to adversary *A*. Challenger answers all the queries of the adversary. At some point, A sends the challenge message m^* and two identities i_0 and i_1. Challenger uniformly chooses one of the identity $b \in \{0,1\}$ and generates the challenge signature $\Sigma^* = (c^*_{v_{i_b}}, c^*_{w_{i_b}}, b_1^*, \Pi^*, sig^*, VK^*, G_3^*)$. Finally, A outputs the bit $b' \in \{0,1\}$.
- *Game* $G_1^{(b)}$: In this experiment, we slightly change *Game* $G_0^{(b)}$ as follows: At the beginning of the game, the challenger generates the one-time signature key pair (VK^*, SK^*) which will be used in the challenge phase. If A requests the opening of a valid signature $\Sigma^* = (c_{v_j}, c_{w_j}, \Pi, VK, G_3, b_1, sig)$ where $VK = VK^*$ the challenger returns a random bit and aborts.
- *Game* $G_2^{(b)}$: In this game, we program the random oracle H_2 in the following way: at the beginning of the game, choose a uniformly random matrix $G_0 \in Z_q^{n \times 2m}$ and set $H_2(VK^*) = G_0$. From the adversary's view, the distribution of G_0 is statistically close to the one in the real attack game, as in [7]. As for other queries, for each fresh H_2 queries on *VK*, the challenger samples small-norm matrices $E_0 \in D^{2m}_{Z^m,\sigma}$ and programs the oracle such that $H_2(VK) = BE_0 \bmod q$. The chosen matrices E_0 are retained for later use.
- *Game* $G_3^{(b)}$: In this game, we program the random oracle H_1 in the following way: At the beginning of the game, a uniform matrix $G_1^* \in Z_q^{n \times m}$ is uniformly chosen and in the challenge phase, set $H_1(cert_{i_b}) = G_1^*$. For other H_1 queries, fresh query on input $cert_i$, challenger samples E_i according to $D^m_{Z^m,\sigma}$ and set $H_1(cert_i) = BE_i \bmod q$. Retain E_i for later use. From the adversary view, the distribution of G_1^*. is same as in *Game* $G_0^{(b)}$.

- *Game* $G_4^{(b)}$: In this game, we modify the way of handling the open and reveal queries. Challenger uniformly chooses a matrix B^* uniformly over $Z_q^{n\times m}$ and to answer any reveal query of user i, it recalls E_i, computes w_i using $cert_i$ and returns as $trace_i$. To answer any open query recall E_0 generated in $Game\ G_2^{(b)}$.
- $Game\ G_5^{(b)}$: In this game, we change the generation of challenge signature. Instead of generating NIZK protocol Π^* using the witness, simulate the protocol and obtain Π'. The challenge signature is $\Sigma^* = (c^*_{v_{i_b}}, c^*_{w_{i_b}}, b_1^*, \Pi', sig^*, VK^*, G_3^*)$. By zero-knowledge protocol of Π^*, the challenge signature generated in this game is computationally indistinguishable from signature generated in $Game\ G_2^{(b)}$
- $Game\ G_6^{(b)}$: We change the way of generating the challenge signature. We modify the generation of challenge cipher texts $(c_1^*, c_2^*, c_3^*, c_4^*)$. Instead of using encryption scheme [7], return random ciphertexts

$$(c_1^*, c_2^*) = (r_1, r_2 + bin(v_{i_b})\tfrac{q}{2}) \quad (22)$$

$$(c_3^*, c_4^*) = (r_3, r_4 + bin(w_{i_b})\frac{q}{2}) \quad (23)$$

Where the vectors (r_1, r_2, r_3, r_4), are uniformly chosen over $\left(Z_q^m \times Z_q^{2m} \times Z_q^n \times Z_q^m\right)$. The challenge signature generated in this game is computationally indistinguishable from $\sum^*$ in $Game\ G_5^{(b)}$ based on the hardness of decision version of LWE assumption.

- *Game* G_7: Finally, we make a slight modification in generation of Σ^*compared to the previous game. Ciphertexts $(c_1^*, c_2^*, c_3^*, c_4^*)$ is uniformly sampled over $\left(Z_q^m \times Z_q^{2m} \times Z_q^n \times Z_q^m\right)$. Signature generated in this game is indistinguishable from the signature in previous game.

Challenge signature Σ^* in the last game is independent of bit $b \in \{0,1\}$. Therefore, advantage of adversary in this game is 0. We proved that the challenge signature generated in game G_7 is computationally indistinguishable from the original anonymity game. Therefore, advantage of adversary in original anonymity game is negligible.

5.3. Framing Attacks

- *Theorem* 3: Our scheme is secure against framing attacks based on the hardness of SIS assumption
- *Proof:* Let A be an adversary that generates a forgery $(m^*,\sum^*)$ which opens to the honest user i^*who did not sign the message m^*. We construct an algorithm B that solves an instance of SIS assumption i.e., given a matrix $\overline{A} \in Z_q^{4n\times 4m}$ as input, algorithm B finds the vector x such that $\overline{A}\,x = 0\ mod\ qand\|x\| \leq 2\beta\sqrt{m}$.

Algorithm B

- *Setup:* Obtain (*gpk,gmsk*) using Setup(1^n) with one modification. Instead of uniformly choosing $F \in Z_q^{4n\times 4m}$, we assign $F = \overline{A}$.

Queries:

- Q_Y: *returns the public-key gpk.*
- Q_S: returns $gmsk$ to A
- $Q_{b\text{-}join}$: A can corrupt the group manager and introduces new user through $Q_{b\text{-}join}$ protocol. At each query, B runs join protocol on behalf on the honest user U_i.
- Q_{Sig} : If A requests for the signature on message m of user I and $i \in U^{(b)}$then, recall ($cert_i$,sec_i) and generate signature using Sign(gpk,sec_i,$cert_i$,m) algorithm.

- *Forgery:* Let A outputs (m^*, Σ^*, τ^*) such that Verify $(gpk, m^*, \sum^*)$=1 with non-negligible probability ε. Let $\Sigma^* = (c_{v^*}, c_{w^*}, \Pi^*, VK^*, G_3^*, b_1^*, sig^*)$.Obtain witness $(j, d_j = [d_{1j}||d_{2j}], z^*, s^*)$ using witness extraction procedure similar to the steps in misidentification attack (coin=1). We consider three cases where A returns 1 in $\mathrm{Exp}^{A}_{fra}(n)$ and show that B solves SIS instance in all these cases.

- $Open(\Sigma^*, gpk, gmsk) = i^* \in U^{(b)}$
- $\forall_{i\in U^{(b)}} Trace\left(\Sigma^*, \mathrm{Reveal}(i)\right) = 1$
- $\forall_{i\in U^{(b)}}(i, \Sigma^*) \in Sigs\ and\ Claim - Verify(\Sigma^*, \tau^*) = 1$

- Case1: Open algorithm decrypts and obtain the vector v_{i^*}. Recall z_{i^*} when answering $Q_{b\text{-}join}$ query such that $Fz_{i^*} = v_{i^*}$. We know $v_{i^*} = Fz^*$. In adversary view, z_{i^*} is chosen according to $D_{\Lambda^{\perp}_{v_{i^*}}(F),\sigma}$, it has atleast n bits of min-entropy. Therefore, x= $z^* - z_{i^*}$ is a solution to SIS instance i.e., $Fx = 0\ mod\ q$and $x \leq 2\beta\sqrt{m}$.
- Case2: $Trace(\Sigma^*, \mathrm{Reveal}(j^*)) = 1$ where $j^* \in U^{(b)}$ then, $D_0 bin(v_{i^*}) + D_1 s_{i^*} = D_0 bin\left(v_{j^*}\right) + D_1 s_{j^*}$ This is possible only if $v_{i^*} = v_{j^*}$ and $s_{i^*} = s_{j^*}$. If $v_{i^*} = v_{j^*}$ then $Fz_{i^*} = Fz_{j^*}$. Therefore, $x = z_{i^*} - z_{j^*}$is a solution to SIS instance.
- Case3: If $\forall_{i\in U^{(b)}}(i, \Sigma^*) \in Sigs\ and\ Claim - Verify(\Sigma^*, \tau^*) = 1$ Let $(j^*, \Sigma^*) \in Sigs$ recall z_{j^*} from $Q_{b\text{-}join}$ protocol such that $v_{j^*} = Fz_{j^*}$. Using improved forking lemma and witness extractor procedure, obtain z^* from τ^* such that $v_{j^*} = Fz^*$. Let $x = z_{j^*} - z^*$ which is a solution to our SIS instance.

6. Conclusions

This work presents the first traceable signature scheme based on lattices. Compared to the existing lattice-based schemes our scheme has additional features like signature claiming and user opening. As agents can run in parallel, user tracing is scalable. Our scheme is based on the work of [12]. Compared to the scheme in [12], our scheme supports signature claiming, user opening and size of *gpk* is efficient by log*N* factor, where *N* is the number of members in the group. Our scheme is proved to be secure based on LWE and SIS assumptions in random oracle model. Construction of lattice-based traceable signature without random-oracle is the future work.

References

[1] Agrawal S., Boneh D., and Boyen X., "Efficient Lattice (H) IBE in the Standard Model," *in Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Monaco and Nice, pp. 553-572, 2010.

[2] Alwen J. and Peikert C., "Generating Shorter Bases for Hard Random Lattices" *Theory of Computing Systems*, vol. 48, no. 3, pp. 535-553, 2011.

[3] Boyen X., "Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More," *in Proceedings of in International Workshop on Public Key Cryptography*, Paris, pp. 499-517, 2010.

[4] Brickell E., Pointcheval D., Vaudenay S., and Yung M., "Design Validations for Discrete Logarithm Based Signature Schemes," *in Proceedings of the International Workshop on Practice and Theory in Public Key Cryptography*, Melbourne, pp. 276-292, 2000.

[5] Chaum D. and Heyst E, "Group Signatures," *in Proceedings of Workshop on the Theory and Application of Cryptographic Techniques*, Brighton, pp. 257-265, 1991.

[6] Fiat A. and Shamir A., "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," *in Proceedings of Conference on the Theory and Application of Cryptographic Techniques*, Santa Barbara, pp. 186-194, 1986.

[7] Gentry C., Peikert C., and Vaikuntanathan V., "Trapdoors for Hard Lattices and New Cryptographic Constructions," *in Proceedings of the fortieth annual ACM Symposium on Theory of Computing*, New York, pp. 197-206, 2008.

[8] Gordon S., Katz J., and Vaikuntanathan V., "A Group Signature Scheme from Lattice Assumptions," *in Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, Singapore, pp. 395-412, 2010.

[9] Kiayias A., Tsiounis Y., and Yung M., "Traceables Signatures," *in Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, pp. 571-589, 2004.

[10] Laguillaumie F., Langlois A., Libert B., and Stehlé D., "Lattice Based Group Signatures with Logarithmic Signature Size," *in Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, Bengaluru, pp. 41-61, 2013.

[11] Langlois A., Ling S., Nguyen K., and Wang H., "Lattice-based Group Signature Scheme with Verifier-Local Revocation," *in Proceedings of International Workshop on Public Key Cryptography*, Buenos Aires, pp. 345-361, 2014.

[12] Libert B., Ling S., Mouhartem F., Nguyen K., and Wang H., "Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions," *in Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, Hanoi, pp. 373-403, 2016.

[13] Libert B., Mouhartem F., and Nguyen K., "A Lattice-Based Group Signature Scheme With Message-Dependent Opening," *in Proceedings of International Conference on Applied Cryptography and Network Security*, London, pp. 137-155, 2016.

[14] Ling S., Nguyen K., Wang H., and Xu Y., "Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease," *in Proceedings of International Conference on Applied Cryptography and Network Security*, Kanazawa, pp. 293-312, 2017.

[15] Regev O., "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1-37, 2009.

[16] Sakthivel A., "Enhancing Cloud Security based on Group Signature," *The International Arab Journal on Information Technology*," vol. 14, no. 6, pp. 923-929, 2017.

[17] Shor P., "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *Society for Industrial and Applied Mathematics Journal*, vol. 41, no. 2 pp. 1484-1509, 1997.

Thakkalapally Preethi Thakkalapally Preethi is pursuing her PhD in Computer Science and Engineering at National Institute of Technology Warangal, India. She received her M Tech in Computer Science (CS) from University of Hyderabad, India in 2014. Her areas of interest are lattice-based cryptography, digital signatures, provable security and algorithms.

Bharat Amberker received his PhD in 1996 from Indian Institute of Science (IISc), Bangalore, India from the Department of Computer Science and Automation. He is presently working as a Professor in Computer Science and Engineering, National Institute of Technology (NIT) Warangal. He is a senior member IEEE, senior member ACM and member of Cryptology Research Society of India. He has guided PhDs in the area of cryptography and security. His research interest includes cryptography, provable security of cryptographic protocols/primitives, algorithms, information security, network security and digital image watermarking.