Receiving $Ch \in \{1,2,3\}$, $\hat{\mathcal{S}}$ replies as follows:

1. If $Ch = 1$, it outputs $\perp$ and aborts.
2. If $Ch = 2$, it sends

$$\begin{aligned}\text{RSP} =&(\{\pi_r, \varphi_r, \phi_r, \boldsymbol{e}''_{i,r} + \boldsymbol{r}'_r, \widehat{\boldsymbol{e}}_{0,r} + \boldsymbol{r}_{0,r}\}_{r=1}^{k}, \tau\\ &\theta_1, \theta_3).\end{aligned}$$

3. If $Ch = 3$, it sends

$$\text{RSP} = (\{\pi_r, \varphi_r, \phi_r, \boldsymbol{r}'_r, \boldsymbol{r}_{0,r}\}_{r=1}^{k}, \tau, \theta_1, \theta_2).$$

–If $\widetilde{Ch} = 2$, $\hat{\mathcal{S}}$ does as follows:

1. Samples the randomness of COM, θ_1, θ_2, θ_3, and several random vectors and permutations,

$$\begin{cases}\boldsymbol{r}'_1, \cdots, \boldsymbol{r}'_k \in \mathbb{Z}_q^{(2\ell+2)3m};\ \boldsymbol{r}_{0,1}, \cdots, \boldsymbol{r}_{0,k} \in \mathbb{Z}_q^{3m};\\ \pi_1, \cdots, \pi_k,\ \varphi_1, \cdots, \varphi_k,\ \phi_1, \cdots, \phi_k \in \mathcal{S}_{3m};\\ \tau \in \mathcal{S}_{2\ell};\ \widehat{\boldsymbol{e}}_{0,1}, \widehat{\boldsymbol{e}}_{0,2}, \cdots, \widehat{\boldsymbol{e}}_{0,k} \in \boldsymbol{B}_{3m};\\ \boldsymbol{id}' \in \boldsymbol{B}_{2\ell};\ \boldsymbol{e}''_{i,1}, \boldsymbol{e}''_{i,2}, \cdots, \boldsymbol{e}''_{i,k} \in \boldsymbol{SecExt}(\boldsymbol{id}').\end{cases}$$

2. Lets $\boldsymbol{r}'_{r,0} = \text{Parse}(\boldsymbol{r}'_r, 1, m)$, and computes $\text{CMT} = (\boldsymbol{c}'_1, \boldsymbol{c}'_2, \boldsymbol{c}'_3)$, where

$$\begin{cases}\boldsymbol{c}'_1 = \text{COM}(\{\pi_r, \varphi_r, \phi_r\}_{r=1}^{k}, \tau, \boldsymbol{A}^* \cdot (\sum_{r=1}^{k} \beta_r \boldsymbol{r}_r),\\ \qquad \boldsymbol{B}^* \cdot (\sum_{r=1}^{k} \beta_r (\boldsymbol{r}'_{r,0}, \boldsymbol{r}_{0,r})); \theta_1),\\ \boldsymbol{c}'_2 = \text{COM}(\{\mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{r}'_r), \phi_r(\boldsymbol{r}_{0,r})\}_{r=1}^{k}; \theta_2),\\ \boldsymbol{c}'_3 = \text{COM}(\{\mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{e}''_{i,r} + \boldsymbol{r}'_r),\\ \qquad \phi_r(\widehat{\boldsymbol{e}}_{0,r} + \boldsymbol{r}_{0,r})\}_{r=1}^{k}; \theta_3).\end{cases}$$

3. Sends CMT to $\mathcal{V}'$.

Receiving $Ch \in \{1,2,3\}$, $\hat{\mathcal{S}}$ replies as follows:

1. If $Ch = 1$, sends

$$\begin{aligned}\text{RSP} =&(\{\mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{e}''_{i,r}), \mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{r}'_r), \phi_r(\widehat{\boldsymbol{e}}_{0,r}),\\ &\phi_r(\boldsymbol{r}_{0,r})\}_{r=1}^{k}, \tau(\boldsymbol{id}'), \theta_2, \theta_3).\end{aligned}$$

2. If $Ch = 2$, it outputs $\perp$ and aborts.
3. If $Ch = 3$, it sends

$$\text{RSP} = (\{\pi_r, \varphi_r, \phi_r, \boldsymbol{r}'_r, \boldsymbol{r}_{0,r}\}_{r=1}^{k}, \tau, \theta_1, \theta_2).$$

–If $\widetilde{Ch} = 3$, $\hat{\mathcal{S}}$ does as follows:

1. Samples the randomness of COM, θ_1, θ_2, θ_3, and several random vectors and permutations,

$$\begin{cases}\boldsymbol{r}'_1, \cdots, \boldsymbol{r}'_k \in \mathbb{Z}_q^{(2\ell+2)3m};\ \boldsymbol{r}_{0,1}, \cdots, \boldsymbol{r}_{0,k} \in \mathbb{Z}_q^{3m};\\ \pi_1, \cdots, \pi_k,\ \varphi_1, \cdots, \varphi_k,\ \phi_1, \cdots, \phi_k \in \mathcal{S}_{3m};\\ \tau \in \mathcal{S}_{2\ell};\ \widehat{\boldsymbol{e}}_{0,1}, \widehat{\boldsymbol{e}}_{0,2}, \cdots, \widehat{\boldsymbol{e}}_{0,k} \in \boldsymbol{B}_{3m};\\ \boldsymbol{id}' \in \boldsymbol{B}_{2\ell};\ \boldsymbol{e}''_{i,1}, \boldsymbol{e}''_{i,2}, \cdots, \boldsymbol{e}''_{i,k} \in \boldsymbol{SecExt}(\boldsymbol{id}').\end{cases}$$

2. Lets $\boldsymbol{e}''_{i,0,r} = \text{Parse}(\boldsymbol{e}''_{i,r}, 1, m)$, $\boldsymbol{r}'_{r,0} = \text{Parse}(\boldsymbol{r}'_r, 1, m)$, and sets $\text{CMT} = (\boldsymbol{c}'_1, \boldsymbol{c}'_2, \boldsymbol{c}'_3)$,

$$\begin{cases}\boldsymbol{c}'_1 = \text{COM}(\{\pi_r, \varphi_r, \phi_r\}_{r=1}^{k}, \tau,\\ \qquad \boldsymbol{A}^* \cdot (\sum_{r=1}^{k} \beta_r (\boldsymbol{e}''_{i,r} + \boldsymbol{r}'_r)) - \boldsymbol{u},\\ \qquad \boldsymbol{B}^* \cdot (\sum_{r=1}^{k} \beta_r (\boldsymbol{e}''_{i,0,r} + \boldsymbol{r}'_{r,0}, \widehat{\boldsymbol{e}}_{0,r} + \boldsymbol{r}_{0,r}))\\ \qquad -\boldsymbol{b}_j; \theta_1),\\ \boldsymbol{c}'_2 = \text{COM}(\{\mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{r}'_r), \phi_j(\boldsymbol{r}_{0,r})\}_{r=1}^{k}; \theta_2),\\ \boldsymbol{c}'_3 = \text{COM}(\{\mathcal{F}_{\pi_j,\varphi_r,\tau}(\boldsymbol{e}''_{i,r} + \boldsymbol{r}'_r),\\ \qquad \phi_r(\widehat{\boldsymbol{e}}_{0,r} + \boldsymbol{r}_{0,r})\}_{r=1}^{k}; \theta_3).\end{cases}$$

3. Sends CMT to $\mathcal{V}'$.

Receiving $Ch \in \{1,2,3\}$, $\hat{\mathcal{S}}$ replies as follows:

1. If $Ch = 1$, it sends

$$\begin{aligned}\text{RSP} =&(\{\mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{e}''_{i,r}), \mathcal{F}_{\pi_r,\varphi_r,\tau}(\boldsymbol{r}'_r), \phi_r(\widehat{\boldsymbol{e}}_{0,r}),\\ &\phi_r(\boldsymbol{r}_{0,r})\}_{r=1}^{k}, \tau(\boldsymbol{id}'), \theta_2, \theta_3).\end{aligned}$$

2. If $Ch = 2$, it sends

$$\begin{aligned}\text{RSP} =&(\{\pi_r, \varphi_r, \phi_r, \boldsymbol{e}''_{i,r} + \boldsymbol{r}'_r, \widehat{\boldsymbol{e}}_{0,r} + \boldsymbol{r}_{0,r}\}_{r=1}^{k},\\ &\tau, \theta_1, \theta_3).\end{aligned}$$

3. If $Ch = 3$, it outputs $\perp$ and aborts.

Based on a statistically hiding property of COM, the distributions of CMT, Ch and RSP are statistically close to those in the real interaction, $\hat{\mathcal{S}}$ outputs $\perp$ and aborts with probability negligibly close to 1/3. Furthermore, once $\hat{\mathcal{S}}$ does not halt, a valid transcript will be given and the distribution of the transcript is statistically close to that in the real interaction, so $\hat{\mathcal{S}}$ can impersonate an honest $\mathcal{P}$ with a probability negligibly close to 2/3.

Argument of knowledge: To prove that the proposed protocol is an argument of knowledge for the relation $\mathcal{R}(n, k, \ell, t, q, m, \beta)$, it needs to prove that the given protocol satisfies the special soundness property.

To show that if there is a $\mathcal{P}'$ (maybe cheating) who can correctly respond to three challenges corresponding to the same commitment CMT with the inputs $\Delta = (\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, j, \boldsymbol{b}_j)$, then there is an extractor $\mathcal{K}$ who can produce

$$(\boldsymbol{id} = \text{Bin}(i), \boldsymbol{e}'_i = (\boldsymbol{e}'_{i,0}, \boldsymbol{e}'_{i,1}, \text{Bin}(i) \otimes \boldsymbol{e}'_{i,1}), \boldsymbol{e}_0)$$

$$s.t.\ (\Delta; \boldsymbol{id} = \text{Bin}(i), \boldsymbol{e}'_i, \boldsymbol{e}_0) \in \mathcal{R}.$$

Indeed, based on three valid $\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3$ given by $\mathcal{P}'$, the extractor $\mathcal{K}$ can extract:

$$\begin{cases} \boldsymbol{t_{id}} \in \boldsymbol{B}_{2\ell}, \boldsymbol{v}'_r \in \boldsymbol{SecExt}(\boldsymbol{t_{id}}), \boldsymbol{v}_r \in \boldsymbol{B}_{3m}, \\ \boldsymbol{c}_1 = \mathrm{COM}(\{\hat{\pi}_r, \hat{\varphi}_r, \hat{\phi}_r\}_{r=1}^k, \hat{\tau}, \boldsymbol{A}^* \cdot (\sum_{r=1}^k \beta_r \boldsymbol{y}'_r) - \\ \qquad \boldsymbol{u}, \boldsymbol{B}^* \cdot (\sum_{r=1}^k \beta_r (\boldsymbol{y}'_{r,0}, \boldsymbol{y}_r)) - \boldsymbol{b}_j; \theta_1) \\ \quad = \mathrm{COM}(\{\tilde{\pi}_r, \tilde{\varphi}_r, \tilde{\phi}_r\}_{r=1}^k, \tilde{\tau}, \boldsymbol{A}^* \cdot (\sum_{r=1}^k \beta_r \boldsymbol{h}'_r), \\ \qquad \boldsymbol{B}^* \cdot (\sum_{r=1}^k \beta_r (\boldsymbol{h}'_{r,0}, \boldsymbol{h}_r)); \theta_1), \\ \boldsymbol{c}_2 = \mathrm{COM}(\{\boldsymbol{w}'_r, \boldsymbol{w}_r\}_{r=1}^k; \theta_2) \\ \quad = \mathrm{COM}(\{\mathcal{F}_{\tilde{\pi}_r, \tilde{\varphi}_r, \tilde{\tau}}(\boldsymbol{h}'_r), \tilde{\phi}_r(\boldsymbol{h}_r)\}_{r=1}^k; \theta_2), \\ \boldsymbol{c}_3 = \mathrm{COM}(\{\boldsymbol{v}'_r + \boldsymbol{w}'_r, \boldsymbol{v}_r + \boldsymbol{w}_r\}_{r=1}^k; \theta_3) \\ \quad = \mathrm{COM}(\{\mathcal{F}_{\hat{\pi}_r, \hat{\varphi}_r, \hat{\tau}}(\boldsymbol{y}'_r), \hat{\phi}_r(\boldsymbol{y}_r)\}_{r=1}^k; \theta_3). \end{cases}$$

Based on the computationally binding property of COM, the extractor $\mathcal{K}$ can deduce that:

$$\begin{cases} \boldsymbol{t_{id}} \in \boldsymbol{B}_{2\ell}, \hat{\tau} = \tilde{\tau}, \hat{\phi}_r = \tilde{\phi}_r, \hat{\pi}_r = \tilde{\pi}_r, \hat{\varphi}_r = \tilde{\varphi}_r; \\ \boldsymbol{A}^* \cdot (\sum_{r=1}^k \beta_r \boldsymbol{y}'_r) - \boldsymbol{u} = \boldsymbol{A}^* \cdot (\sum_{r=1}^k \beta_r \boldsymbol{h}'_r) \bmod q; \\ \boldsymbol{B}^* \cdot (\sum_{r=1}^k \beta_r (\boldsymbol{y}'_{r,0}, \boldsymbol{y}_r)) - \boldsymbol{b}_j \\ \qquad = \boldsymbol{B}^* \cdot (\sum_{r=1}^k \beta_r (\boldsymbol{h}'_{r,0}, \boldsymbol{h}_r)) \bmod q; \\ \boldsymbol{w}'_r = \mathcal{F}_{\tilde{\pi}_r, \tilde{\varphi}_r, \tilde{\tau}}(\boldsymbol{h}'_r), \boldsymbol{v}'_r + \boldsymbol{w}'_r = \mathcal{F}_{\hat{\pi}_r, \hat{\varphi}_r, \hat{\tau}}(\boldsymbol{y}'_r); \\ \boldsymbol{v}'_r \in \boldsymbol{SecExt}(\boldsymbol{t_{id}}), \boldsymbol{v}_r \in \boldsymbol{B}_{3m}; \\ \boldsymbol{w}_r = \tilde{\phi}_r(\boldsymbol{h}_r), \boldsymbol{v}_r + \boldsymbol{w}_r = \hat{\phi}_r(\boldsymbol{y}_r). \end{cases}$$

Let $\boldsymbol{e}'_{i,r} = \boldsymbol{y}'_r - \boldsymbol{h}'_r = \mathcal{T}^{-1}_{\tilde{\pi}_r, \tilde{\varphi}_r, \tilde{\tau}}(\boldsymbol{v}'_r)$, $\boldsymbol{e}_{0,r} = \boldsymbol{y}_r - \boldsymbol{h}_r = \tilde{\phi}_r^{-1}(\boldsymbol{v}_r)$, thus, $\boldsymbol{e}'_{i,r} \in \boldsymbol{SecExt}(\tilde{\tau}^{-1}(\boldsymbol{t_{id}}) = \boldsymbol{id}^*)$, $\boldsymbol{e}_{0,r} \in \boldsymbol{B}_{3m}$. Let $\boldsymbol{e}'_{i,0,r} = \mathrm{Parse}(\boldsymbol{e}'_{i,r}, 1, m)$,

$$\begin{cases} \boldsymbol{A}^* \cdot (\sum_{r=1}^k \beta_r \boldsymbol{e}'_{i,r}) = \boldsymbol{u} \bmod q, \\ \boldsymbol{B}^* \cdot (\sum_{r=1}^k \beta_r (\boldsymbol{e}'_{i,0,r}, \boldsymbol{e}_{0,r})) = \boldsymbol{b}_j \bmod q. \end{cases}$$

The extractor $\mathcal{K}$ produces $\boldsymbol{id} = \mathrm{Bin}(i) \in \{0,1\}^\ell$, $\boldsymbol{e}'_i \in \boldsymbol{Sec}_\beta(\boldsymbol{id})$, and $\boldsymbol{e}_0 \in \mathbb{Z}^m$ as follows:

1. Let $\boldsymbol{id}^* = (d_1, d_2, \cdots, d_\ell, \cdots, d_{2\ell}) = \tilde{\tau}^{-1}(\boldsymbol{t_{id}})$. We obtain $\mathrm{Bin}(i) = \boldsymbol{id} = (d_1, d_2, \cdots, d_\ell)$, and the index $i = \boldsymbol{g}_\ell^\top \cdot \mathrm{Bin}(i)$.

2. Let $\boldsymbol{e}^*_i = \sum_{r=1}^k \beta_r \boldsymbol{e}'_{i,r}$, thus,

$$0 < \|\boldsymbol{e}^*_i\|_\infty \le \sum_{r=1}^k \beta_r \|\boldsymbol{e}'_{i,r}\|_\infty \le \beta.$$

Since $\boldsymbol{e}'_{i,r} \in \boldsymbol{SecExt}(\boldsymbol{id}^*)$, there are two vectors $\boldsymbol{e}^*_{i,0}, \boldsymbol{e}^*_{i,1} \in \mathbb{Z}^{3m}$ that satisfy $\|\boldsymbol{e}^*_{i,0}\|_\infty, \|\boldsymbol{e}^*_{i,1}\|_\infty \le \beta$, and $\boldsymbol{e}^*_i = (\boldsymbol{e}^*_{i,0}, \boldsymbol{e}^*_{i,1}, d_1 \boldsymbol{e}^*_{i,1}, \cdots, d_{2\ell} \boldsymbol{e}^*_{i,1})$. Set

$$\begin{aligned} \boldsymbol{e}'_i &= (\boldsymbol{e}'_{i,0}, \boldsymbol{e}'_{i,1}, d_1 \boldsymbol{e}'_{i,1}, \cdots, d_\ell \boldsymbol{e}'_{i,1}) \\ &= (\boldsymbol{e}'_{i,0}, \boldsymbol{e}'_{i,1}, \mathrm{Bin}(i) \otimes \boldsymbol{e}'_{i,1}), \end{aligned}$$

where $\boldsymbol{e}'_{i,0}$, $\boldsymbol{e}'_{i,1}$ are obtained from $\boldsymbol{e}^*_{i,0}$, $\boldsymbol{e}^*_{i,1}$ by removing the last $2m$ coordinates. Thus, $\boldsymbol{e}'_i \in \boldsymbol{Sec}_\beta(\boldsymbol{id})$ and

$$[\boldsymbol{A}|\boldsymbol{A}_0|\boldsymbol{g}_\ell \otimes \boldsymbol{A}_1] \cdot (\boldsymbol{e}'_{i,0}, \boldsymbol{e}'_{i,1}, \mathrm{Bin}(i) \otimes \boldsymbol{e}'_{i,1}) = \boldsymbol{u} \bmod q.$$

3. Let $\hat{\boldsymbol{e}}_0 = \sum_{r=1}^k \beta_r \boldsymbol{e}_{0,r}$, thus,

$$0 < \|\hat{\boldsymbol{e}}_0\|_\infty \le \sum_{r=1}^k \beta_r \|\boldsymbol{e}_{0,r}\|_\infty \le \beta.$$

Let $\boldsymbol{e}_0 \in \mathbb{Z}^m$ be obtained from $\hat{\boldsymbol{e}}_0$ by removing the last $2m$ coordinates. Thus, $\boldsymbol{e}_0 \in \mathbb{Z}^m$, $0 < \|\boldsymbol{e}_0\|_\infty \le \beta$ and $\boldsymbol{b}_j = (\boldsymbol{B}^\top \hat{\boldsymbol{B}}_j) \cdot \boldsymbol{e}'_i + \boldsymbol{e}_0 \bmod q$.

Finally, $\mathcal{K}$ outputs

$$\mathrm{witness} = (\mathrm{Bin}(i) = \boldsymbol{id}, \boldsymbol{e}'_i \in \boldsymbol{Sec}_\beta(\boldsymbol{id}), \boldsymbol{e}_0 \in \mathbb{Z}^m),$$

which is a valid witness for $\mathcal{R} = (n, k, \ell, t, m, \beta, p, t)$.

4 The VLR-GS-BU scheme

In this section, we describe a lattice-based VLR-GS-BU scheme, and prove the construction satisfying three requirements: correctness, BU-anonymity, and traceability, as defined in Section 2.1. The parameters will also be specified.

4.1 Description of the scheme

−KeyGen$(1^n, N, t)$: Input a parameter n, group size $N = 2^\ell = \mathrm{poly}(n)$, and number of periods $t = \mathrm{poly}(n)$, other parameters are listed in Table 2. This algorithm works as follows:

1. Run TrapGen(q, n, m) to obtain $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor $\boldsymbol{R_A}$.

2. Choose matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, and a vector $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^n$.

3. As in Zhang et al. (2019a, 2019b), for $\boldsymbol{id}$ with $i \in \{0, 1, \cdots, N-1\}$, let $\boldsymbol{A_{id}} = [\boldsymbol{A}|\boldsymbol{A}_0 + i\boldsymbol{A}_1] \in \mathbb{Z}_q^{n \times 2m}$, and proceed as follows:

   3.1. Choose $\boldsymbol{e}_{i,1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m, s}$, let $\boldsymbol{u}_i = (\boldsymbol{A}_0 + i\boldsymbol{A}_1) \cdot \boldsymbol{e}_{i,1}$, and run SamplePre$(\boldsymbol{A}, \boldsymbol{R_A}, \boldsymbol{u} - \boldsymbol{u}_i, s)$ to obtain $\boldsymbol{e}_{i,0} \in \mathbb{Z}^m$.

   3.2. Let $\boldsymbol{e}_i = (\boldsymbol{e}_{i,0}, \boldsymbol{e}_{i,1}) \in \mathbb{Z}^{2m}$, thus $\boldsymbol{A_{id}} \cdot \boldsymbol{e}_i = \boldsymbol{u} \bmod q$ and $0 < \|\boldsymbol{e}_i\|_\infty \le \beta$.

   3.3. For time-period $\mathrm{TP}_{j \in \{1,2,\cdots,t\}} \in \mathbb{Z}_q^n$, define $\boldsymbol{grt}_{i,j} = (\boldsymbol{B}_0 + \mathcal{H}_1(\mathrm{TP}_j) \cdot \boldsymbol{B}_1) \cdot \boldsymbol{e}_{i,0} \bmod q$.

3.4. Let the signing secret-key of member $\boldsymbol{id}$ be $\boldsymbol{gsk}_i = \boldsymbol{e}_i \in \mathbb{Z}^{2m}$, and the revocation token be $\boldsymbol{grt}_i = \{\boldsymbol{grt}_{i,1}, \boldsymbol{grt}_{i,2}, \cdots, \boldsymbol{grt}_{i,t}\}$.

4. Output
Gpk $= (\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \mathcal{G}, \mathcal{H}_1, \mathcal{H}_2)$,
Gsk $= (\boldsymbol{gsk}_0, \boldsymbol{gsk}_1, \cdots, \boldsymbol{gsk}_{N-1})$,
Grt $= (\boldsymbol{grt}_0, \boldsymbol{grt}_1, \cdots, \boldsymbol{grt}_{N-1})$.

−Sign(Gpk, j, $\boldsymbol{gsk}_i$, $\boldsymbol{m}$): Let $\chi \in \mathbb{Z}$ be a β-bounded distribution. Take Gpk, current period j and $\boldsymbol{m} \in \{0,1\}^*$ as inputs, a member $\boldsymbol{id}$ with index i and secret-key $\boldsymbol{gsk}_i = \boldsymbol{e}_i$ proceeds as follows:

1. Choose $\boldsymbol{v} \xleftarrow{\$} \{0,1\}^n$, let
$\boldsymbol{B} = \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \boldsymbol{m}, \boldsymbol{v}) \in \mathbb{Z}_q^{n\times m}$.

2. Choose $\boldsymbol{e}_0 \xleftarrow{\$} \chi^m$, and define
$$\begin{aligned}\boldsymbol{b}_j &= \boldsymbol{B}^\top \cdot \boldsymbol{grt}_{i,j} + \boldsymbol{e}_0 \\ &= (\boldsymbol{B}^\top \cdot (\boldsymbol{B}_0 + \mathcal{H}_1(\mathrm{TP}_j) \cdot \boldsymbol{B}_1)) \cdot \boldsymbol{e}_{i,0} + \boldsymbol{e}_0.\end{aligned}$$

3. Generate a ZKP protocol of which the signer $\boldsymbol{id}$ is a valid member. This is achieved by repeating the protocol in Section 3.2 $\omega(\log n)$ times with $\Delta = (\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{u}, \boldsymbol{B}, \boldsymbol{B}_0, \boldsymbol{B}_1, j, \boldsymbol{b}_j)$ and a witness $(\boldsymbol{id}, \boldsymbol{gsk}_i, \boldsymbol{e}_0)$, and make it non-interactive as $\Pi = (\{\mathrm{CMT}_r\}_{r=1}^{\kappa}, \mathrm{CH}, \{\mathrm{RSP}_r\}_{r=1}^{\kappa})$, where $\mathrm{CH} = \{Ch_r\}_{r=1}^{\kappa} = \mathcal{H}_2(\boldsymbol{m}, \Delta)$.

4. Output $\sigma = (\boldsymbol{m}, j, \Pi, \boldsymbol{v}, \boldsymbol{b}_j)$.

−Verify(Gpk, j, RL_j, $\boldsymbol{m}$, σ): Input Gpk, a signature σ on $\boldsymbol{m} \in \{0,1\}^*$ and a set of tokens $\mathrm{RL}_j = \{\boldsymbol{grt}_{i',j}, \boldsymbol{grt}_{i',j+1}, \cdots, \boldsymbol{grt}_{i',t}\}_{i' \le N-1} \subseteq$ Grt for time-period j, the verifier proceeds as follows:

1. Parse $\sigma = (\boldsymbol{m}, j, \Pi, \boldsymbol{v}, \boldsymbol{b}_j)$.

2. Compute $\boldsymbol{B} = \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \boldsymbol{m}, \boldsymbol{v})$.

3. If $\mathrm{CH} \neq \mathcal{H}_2(\boldsymbol{m}, \Delta)$, return invalid.

4. For $r = 1$ to κ, run the verification steps of the protocol as in Section 3.2 to check the validity of RSP_r *w.r.t* CMT_r and Ch_r. If any condition does not hold, then return invalid.

5. For each $\boldsymbol{grt}_{i',j} \in \mathrm{RL}_j$, compute $\boldsymbol{e}_{i'} = \boldsymbol{b}_j - \boldsymbol{B}^\top \cdot \boldsymbol{grt}_{i',j} \bmod q$. If there exists an index $i' \le N-1$ such that $\|\boldsymbol{e}_{i'}\|_\infty \le \beta$, then return invalid.

6. Return valid.

4.2 Analysis of the scheme

Efficiency: We first analyze the space complexity of our lattice-based VLR-GS-BU scheme, with respect to security parameter n.

−The Gpk only needs $(\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1)$ and a vector $\boldsymbol{u}$ for identity-encoding, two matrices $(\boldsymbol{B}_0, \boldsymbol{B}_1)$ and an FRD function $\mathcal{H}_1$ for the new RT design, and two hash functions $\mathcal{G}$, $\mathcal{H}_2$ modeled as random oracles. So, the bit-size of Gpk is $\mathcal{O}(3nm\log q + 2nm\log q + n\log q) = \widetilde{\mathcal{O}}(n^2)$.

−The member signing secret-key $\boldsymbol{gsk}_i$ is a Gaussian vector $\boldsymbol{e}_i \in \mathbb{Z}^{2m}$ of bit-size $\mathcal{O}(2m) = \widetilde{\mathcal{O}}(n)$.

−The member revocation token $\boldsymbol{grt}_i$ is $t = \mathrm{poly}(n)$ vector $\boldsymbol{grt}_{i,j} \in \mathbb{Z}_q^n$ of bit-size $\mathcal{O}(tn\log q) = \widetilde{\mathcal{O}}(tn)$.

−The signature $\sigma = (\boldsymbol{m}, j, \Pi, \boldsymbol{v}, \boldsymbol{b}_j)$ is of bit-size $\mathcal{O}(\log t + (\ell m \log\beta)\log q + n + m\log q) = \ell \cdot \widetilde{\mathcal{O}}(n)$.

Next, we analyze the computation complexity of our lattice-based VLR-GS-BU scheme *w.s.t* n. Here, we let $r < N$ denote the number of revoked members in the RL and t denote the number of time periods.

−The KeyGen procedure involves one TrapGen operation, for each member, one SamplePre operation, t FRD operations for the vector over $\mathbb{Z}_q^n$, and $t+1$ matrix-vector multiplication operations over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}^m$. Thus, the computation complexity is $\widetilde{\mathcal{O}}(n^2) + N \cdot (\widetilde{\mathcal{O}}(n^2) + t \cdot \widetilde{\mathcal{O}}(n) + (t+1)\cdot\mathcal{O}(n^2)) = N \cdot t \cdot \widetilde{\mathcal{O}}(n^2)$.

−The Sign procedure involves one hash function operation, one matrix-vector multiplication operation, and a proof of the corresponding NIZK in Section 3.2. Thus, the computation complexity is $\widetilde{\mathcal{O}}(n^2) + \mathcal{O}(n^2) + \omega(\log n)\cdot\widetilde{\mathcal{O}}(n^2) = \widetilde{\mathcal{O}}(n^2)$.

−The Verify procedure involves two hash function operations, r matrix-vector multiplication operations, and a verification of the corresponding NIZK in Section 3.2. Thus, the computation complexity is $2\widetilde{\mathcal{O}}(n^2) + r\cdot\mathcal{O}(n^2) + \omega(\log n)\cdot\widetilde{\mathcal{O}}(n^2) = r\cdot\widetilde{\mathcal{O}}(n^2)$.

The detailed comparisons between our construction and previous lattice-based VLR-GS schemes, in terms of asymptotic efficiency, functionality and security, are given in Table 3 and Figs. 1 and 2.

Thus, our lattice-based VLR-GS enjoys better asymptotic efficiency (except a relatively high cost for our KeyGen procedure involving extra t FRD and t matrix-vector multiplication operations). Specifically, we achieve BU security for the first time.

For the correctness, BU-anonymity, traceabil-

Table 3 Comparisons of known lattice-based VLR-GS schemes ($N = 2^\ell$)

Scheme	|Gpk|	$\|\boldsymbol{gsk}\|$	$\|\sigma\|$	Functionality	Free of encryption	BU-security
Ling et al. (2014)	$\ell \cdot \widetilde{\mathcal{O}}(n^2)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	yes	no
Zhang et al. (2016)	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n+\ell)$	VLR	no	no
Gao et al. (2017)	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n+\ell)$	VLR	no	no
Ling et al. (2018)	$\ell \cdot \widetilde{\mathcal{O}}(n^2)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	yes	no
Perera et al. (2018a)	$\ell \cdot \widetilde{\mathcal{O}}(n^2)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	yes	no
Perera et al. (2018b)	$\ell \cdot \widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	fully-dynamic	no	no
Perera et al. (2018c)	$\ell \cdot \widetilde{\mathcal{O}}(n^2)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	fully-dynamic	no	no
Zhang et al. (2019a)	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	yes	no
Zhang et al. (2019b)	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	no	no
Our	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$	$\ell \cdot \widetilde{\mathcal{O}}(n)$	VLR	yes	yes

|Gpk|: the size of group public-key; $|\boldsymbol{gsk}|$: the size of a member's signing secret-key; $|\sigma|$: the size of signature

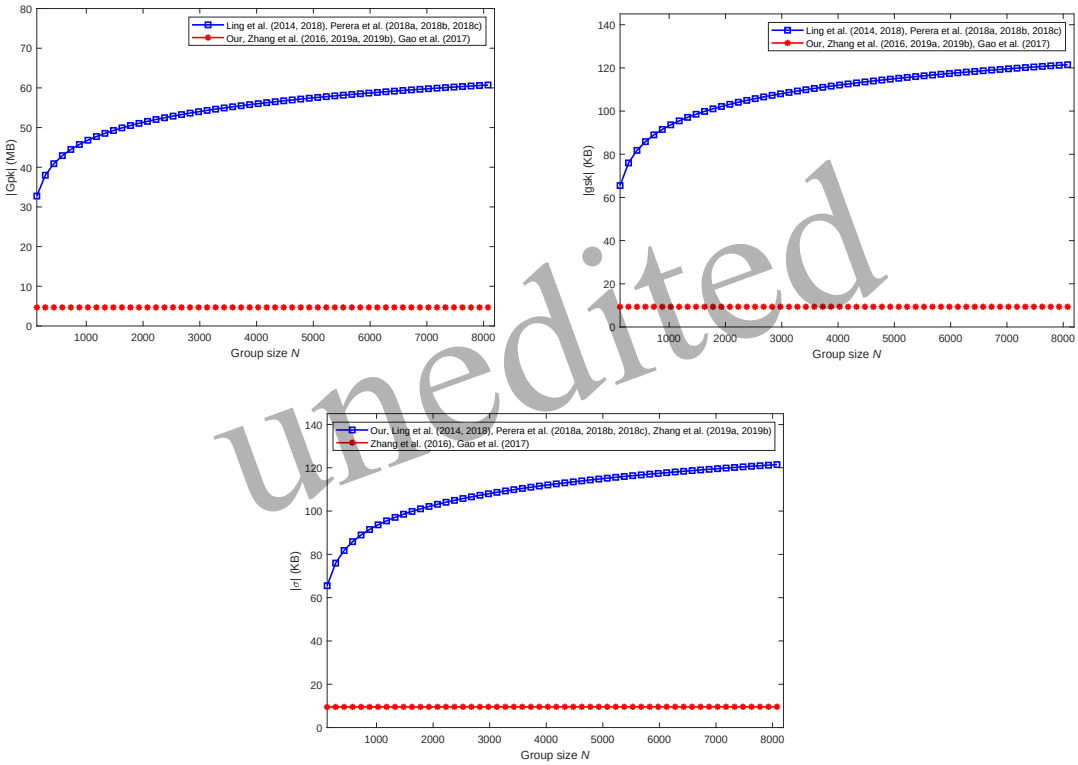

Fig. 1 Comparison of these ten schemes for |Gpk|, $|\boldsymbol{gsk}|$, and $|\sigma|$.

ity, we show the following three theorems and proof details are given in Appendix.

Theorem 2 The proposed scheme is correct with overwhelming probability.

Theorem 3 If COM enjoys the statistically hiding property, the proposed scheme is BU-anonymous in the random oracle model.

Theorem 4 If the $\mathrm{SIS}^{\infty}_{n,m,q,2\beta\cdot(1+\omega(\sqrt{\log m}))}$ problem is hard, then the proposed scheme is traceable in the random oracle model.

5 Conclusion

In this paper, we proposed the first lattice-based VLR-GS scheme with BU security, and thus resolved a prominent open problem. By creatively adopting an injective encoding function with FRD, a compact identity-encoding technique and the corresponding Stern-type statistical ZKP protocol, our new scheme enjoys a $\mathcal{O}(\log N)$ factor savings for bit-sizes of GPK and member's signing secret-key, and is free of any public-key encryption. Moreover, with BU security,

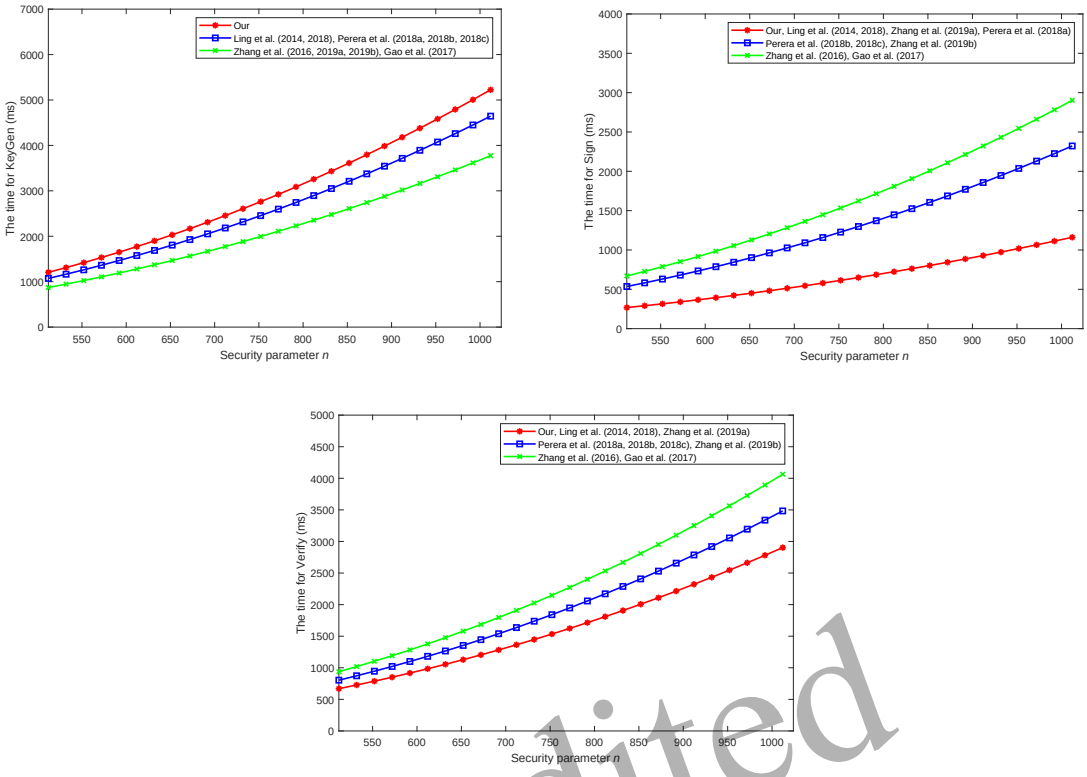

Fig. 2 Comparison of these ten schemes for KeyGen cost, Sign cost, and Verify cost.

it is more suitable for some large group with better security.

Contributors

Yanhua ZHANG and Huiwen JIA designed the research. Yanhua ZHANG processed the data and drafted the manuscript. Ximeng LIU helped organize the manuscript. Yupu HU and Yong GAN revised and finalized the paper.

Compliance with ethics guidelines

Yanhua ZHANG, Ximeng LIU, Yupu HU, Yong GAN and Huiwen JIA declare that they have no conflict of interest.

References

Agrawal S, Boneh D, Boyen X, 2010. Efficient lattice (H)IBE in the standard model. 29th Int Conf on the Theory and Applications of Cryptographic Techniques, p.553-572. https://doi.org/10.1007/978-3-642-13190-5_28

Ajtai M, 1996. Generating hard instances of lattice problems (extended abstract). 28th ACM Symp on Theory of Computing, p.99-108. https://doi.org/10.1145/237814.237838

Alwen J, Peikert C, 2011. Generating shorter bases for hard random lattices. *Theor Comput Sci*, 48(3):535-553. https://doi.org/10.1007/s00224-010-9278-3

Bellare M, Micciancio D, Warinschi B, 2003. Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. 22nd Int Conf on the Theory and Applications of Cryptographic Techniques, p.614-629. https://doi.org/10.1007/3-540-39200-9_38

Bellare M, Shi H, Zhang C, 2005. Foundations of group signatures: the case of dynamic groups. Cryptographers' Track at the RSA Conf, p.136-153. https://doi.org/10.1007/978-3-540-30574-3_11

Boneh D, Shacham H, 2004. Group signatures with verifier-local revocation. 11th ACM Conf on Computer and Communications Security, p.168-177. https://doi.org/10.1145/1030083.1030106

Bootle J, Cerulli A, Chaidos P, et al., 2016. Foundations of fully dynamic group signatures. 14th Int Conf on the Applied Cryptography and Network Security, p.117-136. https://doi.org/10.1007/978-3-319-39555-5_7

Cash D, Hofheinzy D, Kiltz E, et al., 2010. Bonsai trees, or how to delegate a lattice basis. 29th Int Conf on the Theory and Applications of Cryptographic Techniques, p.523-552. https://doi.org/10.1007/978-3-642-13190-5_27

Chaum D, van Heyst E, 1991. Group signatures. Workshop on the Theory and Application of Cryptographic Techniques, p:257-265. https://doi.org/10.1007/3-540-46416-6_22

Emura K, Hayashi T, 2018. A revocable group signature scheme with scalability from simple assumptions and its

implementation. 21st Int Conf on Information Security, p.442-460.
https://doi.org/10.1007/978-3-319-99136-8_24

Gao W, Hu YP, Zhang YH, et al., 2017. Lattice-based group signature with verifier-local revocation. *J Shanghai JiaoTong Univ* (*Sci*) 22(3):313-321.
https://doi.org/10.1007/s12204-017-1837-1

Gentry C, Peikert C, Vaikuntanathan V, 2008. Trapdoor for hard lattices and new cryptographic constructions. 40th ACM Symp on Theory of Computing, p.197-206.
https://doi.org/10.1145/1374376.1374407

Gordon SD, Katz J, Vaikuntanathan V, 2010. A group signature scheme from lattice assumptions. 16th Int Conf on the Theory and Application of Cryptology and Information Security, p.395-412.
https://doi.org/10.1007/978-3-642-17373-8_23

Huang JY, Huang Q, Susilo W, 2020. Leakage-resilient group signature: definitions and constructions. *Inform Sci*, 509(2020):119-132.
https://doi.org/10.1016/j.ins.2019.09.004

Ishida A, Sakai Y, Emura K, et al., 2018. Fully anonymous group signature with verifier-local revocation. 11th Int Conf on Security and Cryptography for Networks, p.23-42.
https://doi.org/10.1007/978-3-319-98113-0_2

Kawachi A, Tanaka K, Xagawa K, 2008. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. 14th Int Conf on the Theory and Application of Cryptology and Information Security, p.372-389.
https://doi.org/10.1007/978-3-540-89255-7_23

Langlois A, Ling S, Nguyen K, et al., 2014. Lattice-based group signature scheme with verifier-local revocation. 17th Int Conf on Practice and Theory in Public-Key Cryptography, p.345-361.
https://doi.org/10.1007/978-3-642-54631-0_20

Libert B, Vergnaud D, 2009. Group signatures with verifier-local revocation and backward unlinkability in the standard model. 8th Int Conf on Cryptology and Network Security, p.498-517.
https://doi.org/10.1007/978-3-642-10433-6_34

Ling S, Nguyen K, Langlois A, et al., 2018. A lattice-based group signature scheme with verifier-local revocation. *Theor Comput Sci*, 730:1-20.
https://doi.org/10.1016/j.tcs.2018.03.027

Ling S, Nguyen K, Stehlé D, et al., 2013. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. 16th Int Conf on Practice and Theory in Public-Key Cryptography, p.107-124.
https://doi.org/10.1007/978-3-642-36362-7_8

Micciancio D, Peikert C, 2013. Hardness of SIS and LWE with small parameters. 33rd Annual Cryptology Conf, p.21-39.
https://doi.org/10.1007/978-3-642-40041-4_2

Micciancio D, Peikert C, 2012. Trapdoors for lattices: simpler, tighter, faster, smaller. 31st Int Conf on the Theory and Applications of Cryptographic Techniques, p.700-718.
https://doi.org/10.1007/978-3-642-29011-4_41

Nakanishi T, Funabiki N, 2005. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. 11th Int Conf on the Theory and Application of Cryptology and Information Security, p.533-548.
https://doi.org/10.1007/11593447_29

Nakanishi T, Funabiki N, 2006. A short verifier-local revocation group signature scheme with backward unlinkability. 1st Int Workshop on Security, p.17-32.
https://doi.org/10.1007/11908739_2

Nguyen PQ, Zhang J, Zhang ZF, 2015. Simpler efficient group signature from lattices. 18th Int Conf on Practice and Theory in Public-Key Cryptography, p.401-426.
https://doi.org/10.1007/978-3-662-46447-2_18

Perera MNS, Koshiba T, 2018a. Achieving full security for lattice-based group signatures with verifier-local revocation. 20th Int Conf on Information and Communications Security, p.287-302.
https://doi.org/10.1007/978-3-030-01950-1_17

Perera MNS, Koshiba T, 2018b. Zero-knowledge proof for lattice-based group signature schemes with verifier-local revocation. 21st Int Conf on Network-Based Information Systems, p.287-302.
https://doi.org/10.1007/978-3-319-98530-5_68

Perera MNS, Koshiba T, 2018c. Achieving strong security and verifier-local revocation for dynamic group signatures from lattice assumptions. 14th Int Conf on Security and Trust Management, p.3-19.
https://doi.org/10.1007/978-3-030-01141-3_1

Regev O, 2005. On lattices, learning with errors, random linear codes, and cryptography. 37th ACM Symp on Theory of computing, p.84-93.
https://doi.org/10.1145/1060590.1060603

Song DX, 2001. Practical forward secure group signature schemes. 8th ACM Conf on Computer and Communications Security, p.225-234.
https://doi.org/10.1145/501983.502015

Zhang YH, Hu YP, Gao W, et al., 2016. Simpler efficient group signature scheme with verifier-local revocation from lattices. *KSII Trans Internet Inf Syst* 10(1):414-430.
https://doi.org/10.3837/tiis.2016.01.024

Zhang YH, Hu YP, Zhang QK, et al., 2019a. On new zero-knowledge proofs for lattice-based group signatures with verifier-local revocation. 22nd Int Conf on Information Security, p.190-208.
https://doi.org/10.1007/978-3-030-30215-3_10

Zhang YH, Liu XM, Hu YP, et al., 2019b. Lattice-based group signatures with verifier-local revocation: achieving shorter key-sizes and explicit traceability with ease. 18th Int Conf on Cryptology and Network Security, p.120-140.
https://doi.org/10.1007/978-3-030-31578-8_7

Appendix. Proofs for the VLR-GS-BU

Proof of Theorem 2.

Proof For the first four steps of Verify, a member $\boldsymbol{id}$ with an identity index i having a valid witness $(\boldsymbol{e}_i', \boldsymbol{e}_0) \in \boldsymbol{Sec}_\beta(\boldsymbol{id}) \times \chi^m$ can return a signature meeting it. As for step 5, $\boldsymbol{e}_{i'}$ can be expressed as,

$$\begin{aligned}\boldsymbol{e}_{i'} &= \boldsymbol{b}_j - \boldsymbol{B}^\top \cdot \boldsymbol{grt}_{i',j} \\ &= \boldsymbol{B}^\top \cdot (\boldsymbol{grt}_{i,j} - \boldsymbol{grt}_{i',j}) + \boldsymbol{e}_0 \bmod q.\end{aligned}$$

1. To prove that $\boldsymbol{grt}_{i,j} \notin \mathrm{RL}_j \Rightarrow \mathrm{Verify}(\cdot) = \mathrm{valid}$.

   Suppose that $\boldsymbol{grt}_{i,j} \notin \mathrm{RL}_j$, to prove that with overwhelming probability, step 5 is satisfied, i.e., $\mathrm{Verify}(\mathrm{Gpk}, j, \mathrm{RL}_j, \mathrm{Sign}(\mathrm{Gpk}, j, \boldsymbol{gsk}_i, \boldsymbol{m}), \boldsymbol{m}) = \mathrm{valid}$ and $||\boldsymbol{e}_{i'}||_\infty > \beta$. For all $\boldsymbol{grt}_{i',j} \in \mathrm{RL}_j$; we have $\boldsymbol{B}^\top \cdot (\boldsymbol{grt}_{i,j} - \boldsymbol{grt}_{i',j}) = \boldsymbol{e}_{i'} - \boldsymbol{e}_0 \bmod q$. Let $\boldsymbol{s}_{i',j} = \boldsymbol{grt}_{i,j} - \boldsymbol{grt}_{i',j}$, we have $\|\boldsymbol{B}^\top \boldsymbol{s}_{i',j}\|_\infty \leq \|\boldsymbol{e}_{i'}\|_\infty + \|\boldsymbol{e}_0\|_\infty \leq \|\boldsymbol{e}_{i'}\|_\infty + \beta$. According to Lemma 4 of Ling et al. (2018),

   $$\Pr[\|\boldsymbol{B}^\top \boldsymbol{s}_{i',j}\|_\infty \leq 2\beta] \leq 1/(4\beta+1)^n.$$

   Thus, $\|\boldsymbol{e}'_i\|_\infty > 2\beta - \beta = \beta$ is satisfied with an overwhelming probability ($> 1 - (4\beta+1)^{-n}$).

2. To prove that $\mathrm{Verify}(\cdot) = \mathrm{valid} \Rightarrow \boldsymbol{grt}_{i,j} \notin \mathrm{RL}_j$.

   Assume $\mathrm{Verify}(\cdot) = \mathrm{valid}$; for all $\boldsymbol{grt}_{i',j} \in \mathrm{RL}_j$, we have $\|\boldsymbol{e}_{i'}\|_\infty > \beta$. If there is an i' satisfying $\boldsymbol{grt}_{i,j} = \boldsymbol{grt}_{i',j}$, we have $\boldsymbol{e}_{i'} = \boldsymbol{e}_0$, $\|\boldsymbol{e}_{i'}\|_\infty = \|\boldsymbol{e}_0\|_\infty \leq \beta$. Thus, a contradiction exists and the above relation holds with probability 1.

Proof of Theorem 3.

Proof A list of games is established as follows:

Game 0. $\mathcal{C}$ honestly proceeds as follows:

1. Run KeyGen to get $(\mathrm{Gpk}, \mathrm{Gsk}, \mathrm{Grt})$. Set $\mathrm{RL} = \varnothing$, $\mathrm{Corr} = \varnothing$, and send Gpk to $\mathcal{A}$.
2. For $\mathcal{A}$'s signing queries on $\boldsymbol{m} \in \{0,1\}^*$ of member $i \leq N-1$ for $j \in \{1, 2, \cdots, t\}$, $\mathcal{C}$ returns $\sigma \leftarrow \mathrm{Sign}(\mathrm{Gpk}, j, \boldsymbol{gsk}_i, \boldsymbol{m})$; for $\mathcal{A}$'s corruption queries for i, $\mathcal{C}$ sets $\mathrm{Corr} = \mathrm{Corr} \cup \{\boldsymbol{id}, i\}$ and returns $\boldsymbol{gsk}_i$; for $\mathcal{A}$'s revocation queries for period j, $\mathcal{C}$ sets $\mathrm{RL} = \mathrm{RL} \cup \{\boldsymbol{grt}_{i,j}\}$ and returns it.
3. $\mathcal{A}$ outputs a message $\boldsymbol{m}^* \in \{0,1\}^*$, a period $j^* \in \{1, 2, \cdots, t\}$, two indices i_0, i_1, and for each $b \in \{0,1\}$, $\boldsymbol{grt}_{i_b,1}, \boldsymbol{grt}_{i_b,2}, \cdots, \boldsymbol{grt}_{i_b,j^*} \notin \mathrm{RL}$.
4. $\mathcal{C}$ picks $b \xleftarrow{\$} \{0,1\}$, generates a signature $\sigma^* = \mathrm{Sign}(\mathrm{Gpk}, j^*, \boldsymbol{gsk}_{i_b}, \boldsymbol{m}^*) = (\boldsymbol{m}^*, j^*, \Pi, \boldsymbol{v}, \boldsymbol{b}_{j^*})$.
5. $\mathcal{A}$ makes queries as before without the right to ask for $\boldsymbol{gsk}_{i_b}$ or $\boldsymbol{grt}_{i_b,j}$ for each $b \in \{0,1\}$ and each $j \in \{1, 2, \cdots, j^*\}$.
6. $\mathcal{A}$ outputs $b' \in \{0,1\}$.

Game 1: $\mathcal{C}$ simulates step 4 of Game 0 by programming the oracle:

1. Choose $\boldsymbol{v} \xleftarrow{\$} \{0,1\}^n$ and $\boldsymbol{e}_0 \xleftarrow{\$} \chi^m$.
2. Define $\boldsymbol{B} = \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \boldsymbol{m}^*, \boldsymbol{v})$ and $\boldsymbol{b}_{j^*} = \boldsymbol{B}^\top \cdot \boldsymbol{grt}_{i_b,j^*} + \boldsymbol{e}_0 \bmod q$.
3. Programs $\mathcal{H}_2$, and Π^* is statistically close to Π.
4. Outputs $\widehat{\sigma}^* = (\boldsymbol{m}^*, j^*, \Pi^*, \boldsymbol{v}, \boldsymbol{b}_{j^*})$.

Game 2: $\mathcal{C}$ defines $\boldsymbol{b}_{j^*} = \boldsymbol{B}^\top \boldsymbol{r} + \boldsymbol{e}_0$, so $\boldsymbol{b}_{j^*}$ is statistically close to the one in Game 1, and thus, Game 2 is statistically indistinguishable from Game 1.

Game 3: $\mathcal{C}$ gets $(\boldsymbol{B}, \boldsymbol{b}_{j^*}) \xleftarrow{\$} \mathcal{U}$, so $(\boldsymbol{B}, \boldsymbol{b}_{j^*})$ is close to the one in Game 2. Games 3 and 2 are computationally indistinguishable. Furthermore, the advantage $\mathrm{Adv}_{\mathcal{A}}^{\text{BU-anon}}$ is 0.

According to the indistinguishability of Games 1, 2, and 3, the advantage $\mathrm{Adv}_{\mathcal{A}}^{\text{BU-anon}}$ in Game 1 is negligible, i.e., our new scheme is BU-anonymous.

Proof of Theorem 4.

Proof Suppose a forger $\mathcal{F}^*$ breaks the scheme with advantage ϵ; using $\mathcal{F}^*$, we design an efficient $\mathcal{A}$ to solve the $\mathrm{SIS}^{\infty}_{n,m,q,2\beta\cdot(1+\omega(\sqrt{\log m}))}$ problem.

Setup: $\mathcal{A}$ proceeds as follows:

1. Choose $i^* \in \{0, 1, \cdots, N-1\}$, $\boldsymbol{e}^*_{i^*,0}$, $\boldsymbol{e}^*_{i^*,1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m, s}$ and $\boldsymbol{R} \xleftarrow{\$} \{-1, 1\}^{m \times m}$.
2. Run TrapGen to obtain $\boldsymbol{A}_1 \in \mathbb{Z}_q^{n \times m}$ and $\boldsymbol{R}_{\boldsymbol{A}_1}$.
3. Define $\boldsymbol{A} = \hat{\boldsymbol{A}}$ and $\boldsymbol{A}_0 = \boldsymbol{A} \cdot \boldsymbol{R} - i^* \boldsymbol{A}_1 \bmod q$.
4. Sample $\boldsymbol{B}_0, \boldsymbol{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and define $\boldsymbol{u} = \boldsymbol{A} \cdot (\boldsymbol{e}^*_{i^*,0} + \boldsymbol{R} \cdot \boldsymbol{e}^*_{i^*,1}) \bmod q$.
5. For period $\mathrm{TP}_{j \in \{1,2,\cdots,t\}}$, define $\boldsymbol{grt}_{i^*,j} = (\boldsymbol{B}_0 + \mathcal{H}_1(\mathrm{TP}_j) \cdot \boldsymbol{B}_1) \cdot \boldsymbol{e}^*_{i^*,0} \bmod q$.
6. For $i = i^*$, let $\boldsymbol{gsk}_{i^*} = (\boldsymbol{e}^*_{i^*,0}, \boldsymbol{e}^*_{i^*,1})$ and $\boldsymbol{grt}_{i^*} = \{\boldsymbol{grt}_{i^*,1}, \boldsymbol{grt}_{i^*,2}, \cdots, \boldsymbol{grt}_{i^*,t}\}$.
7. For $i \neq i^*$, define $\boldsymbol{A}_{\boldsymbol{id}} = [\boldsymbol{A} | \boldsymbol{A}_0 + i\boldsymbol{A}_1]$ and run $\mathrm{SampleRight}(\boldsymbol{A}, (i-i^*)\boldsymbol{A}_1, \boldsymbol{R}, \boldsymbol{R}_{\boldsymbol{A}_1}, \boldsymbol{u}, s)$ to get $\boldsymbol{e}_i = (\boldsymbol{e}_{i,0}, \boldsymbol{e}_{i,1}) \in \mathbb{Z}^{2m}$. Then let $\boldsymbol{gsk}_i = \boldsymbol{e}_i$ and $\boldsymbol{grt}_i = \{\boldsymbol{grt}_{i,1}, \boldsymbol{grt}_{i,2}, \cdots, \boldsymbol{grt}_{i,t}\}$, where $\boldsymbol{grt}_{i,j} = (\boldsymbol{B}_0 + \mathcal{H}_1(\mathrm{TP}_j) \cdot \boldsymbol{B}_1) \cdot \boldsymbol{e}_{i,0} \bmod q$.
8. Let $\mathcal{H}_1 : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ be an FRD function, $\mathcal{H}_2 : \{0,1\}^* \to \{1,2,3\}^{\kappa = \omega(\log n)}$ and $\mathcal{G} : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ be two hash functions.

9. Let $\mathrm{Gpk} = (\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{G})$, $\mathrm{Gsk} = (\boldsymbol{gsk}_0, \boldsymbol{gsk}_1, \cdots, \boldsymbol{gsk}_{N-1})$ and $\mathrm{Grt} = (\boldsymbol{grt}_0, \cdots, \boldsymbol{grt}_{N-1})$, send $(\mathrm{Gpk}, \mathrm{Grt})$ to $\mathcal{F}^*$.

Queries: $\mathcal{F}^*$ proceeds as follows:

1. Corrupting: Take $\boldsymbol{id}$ with an index i as input, $\mathcal{A}$ outputs $\boldsymbol{gsk}_i$ and adds $(\boldsymbol{id}, i)$ to Corr.
2. Signing: Take $\boldsymbol{m} \in \{0,1\}^*$ of i at period j as input, $\mathcal{A}$ outputs $\sigma \leftarrow \mathrm{Sign}(\mathrm{Gpk}, j, \boldsymbol{gsk}_i, \boldsymbol{m})$. In particular, the values in $\{1,2,3\}^{\kappa=\omega(\log n)}$ are sampled as responses to $\mathcal{H}_2$. Let r_d be a reply to the d-th ($d \le q_{\mathcal{H}_2}$) query; here, $q_{\mathcal{H}_2}$ is the whole number of queries to $\mathcal{H}_2$.

Forgery: $\mathcal{F}^*$ returns $\boldsymbol{m}^* \in \{0,1\}^*$, $\mathrm{RL}^*_{j^*} \subseteq \mathrm{Grt}$ for period j^* and a forged $\sigma^* = (\boldsymbol{m}^*, j^*, \Pi^*, \boldsymbol{v}^*, \boldsymbol{b}^*_{j^*})$, which satisfies the following:

1. $\mathrm{Verify}(\mathrm{Gpk}, j^*, \mathrm{RL}^*_{j^*}, \sigma^*, \boldsymbol{m}^*) = \mathrm{valid}$.
2. The implicit-tracing does not succeed, or returns a member not included in $\mathrm{Corr}\backslash\mathrm{RL}^*_{j^*}$.
3. $\mathcal{A}$ has not obtained σ^* by a signing query.

$\mathcal{F}^*$ proceeds as in Zhang et al. (2019 b) and let $\boldsymbol{B}^* = \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{u}, \boldsymbol{m}^*, \boldsymbol{v}^*)$. $\mathcal{A}$ obtains a 3-fork involving

$$(\boldsymbol{m}^*, \boldsymbol{A}, \boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{u}, \boldsymbol{B}^*, \boldsymbol{B}_0, \boldsymbol{B}_1, j^*, \boldsymbol{b}^*_{j^*}, \mathrm{CMT}_r\}_{r=1}^{\kappa})$$

after at most $32 \cdot q_{\mathcal{H}_2}/(\epsilon - 3^{-\kappa})$ executions of $\mathcal{F}^*$. With the help of an extractor $\mathcal{K}$ as described in the Argument of Knowledge, we get a valid witness $= (\boldsymbol{id} = \mathrm{Bin}(i) \in \{0,1\}^{\ell}, \boldsymbol{e}_i = (\boldsymbol{e}_{i,0}, \boldsymbol{e}_{i,1}) \in \mathbb{Z}^{2m}, \boldsymbol{e}_0 \in \mathbb{Z}^m)$ such that,

1. $[\boldsymbol{A}|\boldsymbol{A}_0 + i\boldsymbol{A}_1] \cdot \boldsymbol{e}_i = \boldsymbol{u} \bmod q$, $\boldsymbol{e}_i \in \boldsymbol{Sec}_\beta(\boldsymbol{id})$.
2. $\boldsymbol{b}^*_{j^*} = (\boldsymbol{B}^{*\top} \cdot \hat{\boldsymbol{B}_{j^*}}) \cdot \boldsymbol{e}_{i,0} + \boldsymbol{e}^*_0$, $0 < \|\boldsymbol{e}_0\|_\infty \le \beta$.

Thus, we show two cases:

1. If $i \neq i^*$ (the probability is at most $1 - 1/N$), $\mathcal{A}$ aborts.
2. If $i = i^*$, $\mathcal{A}$ returns $\hat{\boldsymbol{e}} = (\boldsymbol{e}^*_{i^*,0} - \boldsymbol{e}_{i^*,0}) + \boldsymbol{R} \cdot (\boldsymbol{e}^*_{i^*,1} - \boldsymbol{e}_{i^*,1})$. Thus, we have

$$\begin{aligned}\hat{\boldsymbol{A}} \cdot \hat{\boldsymbol{e}} &= \boldsymbol{A}(\boldsymbol{e}^*_{i^*,0} - \boldsymbol{e}_{i^*,0} + \boldsymbol{R}(\boldsymbol{e}^*_{i^*,1} - \boldsymbol{e}_{i^*,1})) \\ &= \underbrace{\boldsymbol{A}(\boldsymbol{e}^*_{i^*,0} + \boldsymbol{R}\boldsymbol{e}^*_{i^*,1})}_{\boldsymbol{u}} - \underbrace{\boldsymbol{A}(\boldsymbol{e}_{i^*,0} + \boldsymbol{R}\boldsymbol{e}_{i^*,1})}_{\boldsymbol{u}} \\ &= \boldsymbol{0} \bmod q.\end{aligned}$$

We now show that with a high probability, $\hat{\boldsymbol{e}} \neq \boldsymbol{0} \bmod q$ and $\|\hat{\boldsymbol{e}}\| \le \mathrm{poly}(m)$:

1. $\|\hat{\boldsymbol{e}}\|_\infty \le \mathrm{poly}(m)$. For $b \in \{0,1\}$, $\|\boldsymbol{e}^*_{i^*,b}\|_\infty \le \beta$, $\|\boldsymbol{e}_{i^*,b}\|_\infty \le \beta$, $\boldsymbol{R} \xleftarrow{\$} \{1,-1\}^{m \times m}$, thus, we have $\|\hat{\boldsymbol{e}}\|_\infty \le (1 + \omega(\sqrt{\log m})) \cdot 2\beta = \mathrm{poly}(m)$.
2. $\hat{\boldsymbol{e}} \neq \boldsymbol{0} \bmod q$. Since $\sigma^* = (\boldsymbol{m}^*, j^*, \Pi^*, \boldsymbol{v}^*, \boldsymbol{b}^*_{j^*})$ is a forged signature, the implicit-tracing does not succeed, or returns a member not included in $\mathrm{Corr}\backslash\mathrm{RL}^*_{j^*}$.
   2.1. If the implicit-tracing will not succeed, then $\mathrm{Verify}(\mathrm{Gpk}, j^*, \boldsymbol{grt}_{i^*,j^*}, \sigma^*, \boldsymbol{m}^*) = \mathrm{valid}$ implies that $\hat{\boldsymbol{B}_{j^*}} \cdot \boldsymbol{e}_{i^*,0} \bmod q \neq \boldsymbol{grt}_{i^*,j^*} = \hat{\boldsymbol{B}_{j^*}} \cdot \boldsymbol{e}^*_{i^*,0} \bmod q$, thus, $\boldsymbol{e}_{i^*,0} \neq \boldsymbol{e}^*_{i^*,0}$.
   2.2. If the implicit-tracing algorithm traces to a member $\hat{i}^* \notin \mathrm{Corr}\backslash\mathrm{RL}^*_{j^*}$, clearly, we have the following facts:

$$\begin{cases}\mathrm{Verify}(\mathrm{Gpk}, j^*, \boldsymbol{grt}_{\hat{i}^*,j^*}, \sigma^*, \boldsymbol{m}^*) = \mathrm{invalid}, \\ \mathrm{Verify}(\mathrm{Gpk}, j^*, \mathrm{RL}^*_{j^*}, \sigma^*, \boldsymbol{m}^*) = \mathrm{valid}.\end{cases}$$

Thus, we have that:

   2.2.1. $\boldsymbol{grt}_{\hat{i}^*,j^*} \notin \mathrm{RL}^*_{j^*}$, thus, $\hat{i}^* \notin \mathrm{Corr}$.
   2.2.2. Since $\|\boldsymbol{b}^*_{j^*} - \boldsymbol{B}^{*\top} \cdot \boldsymbol{grt}_{\hat{i}^*,j^*}\|_\infty = \|\boldsymbol{B}^{*\top} \cdot (\hat{\boldsymbol{B}_{j^*}} \cdot \boldsymbol{e}_{i^*,0} - \boldsymbol{grt}_{\hat{i}^*,j^*}) + \boldsymbol{e}_0\|_\infty \le \beta$, $\|\boldsymbol{e}_0\|_\infty \le \beta$, thus $\|\boldsymbol{B}^{*\top} \cdot (\hat{\boldsymbol{B}_{j^*}} \cdot \boldsymbol{e}_{i^*,0} - \boldsymbol{grt}_{\hat{i}^*,j^*})\|_\infty \le 2\beta$. Further, according to Lemma 4 of Ling et al. (2018), we have that $\boldsymbol{grt}_{\hat{i}^*,j^*} = \hat{\boldsymbol{B}_{j^*}} \cdot \boldsymbol{e}_{i^*,0} \bmod q$ with overwhelming probability.

Now, consider the following two cases:

   2.2.3. If $\mathcal{F}^*$ did not request $\boldsymbol{gsk}_{i^*}$, then vector $(\boldsymbol{e}^*_{i^*,0}, \boldsymbol{e}^*_{i^*,1})$ cannot be known to $\mathcal{F}^*$, thus, according to Lemma 1, we have that $(\boldsymbol{e}^*_{i^*,0}, \boldsymbol{e}^*_{i^*,1}) \neq (\boldsymbol{e}_{i^*,0}, \boldsymbol{e}_{i^*,1})$ with overwhelming probability.
   2.2.4. If $\mathcal{F}^*$ requested $\boldsymbol{gsk}_{i^*}$, then $i^* \in \mathrm{Corr}$, thus $i^* \neq \hat{i}^*$, therefore, $\boldsymbol{grt}_{i^*,j^*} \neq \boldsymbol{grt}_{\hat{i}^*,j^*}$, which means $\boldsymbol{e}_{i^*,0} \neq \boldsymbol{e}^*_{i^*,0}$.

The following analysis is same as in Zhang et al. (2019b). For the different cases in 2.1 and 2.2.4 (assume that $\boldsymbol{e}^*_{i^*,1} = \boldsymbol{e}_{i^*,1}$), and in 2.1, 2.2.3 and 2.2.4 (assume that $\boldsymbol{e}^*_{i^*,1} \neq \boldsymbol{e}_{i^*,1}$), we conclude that with probability $1 - \exp^{-\widetilde{\mathcal{O}}(n)}$, $\hat{\boldsymbol{e}} = (\boldsymbol{e}^*_{i^*,0} - \boldsymbol{e}_{i^*,0}) + \boldsymbol{R} \cdot (\boldsymbol{e}^*_{i^*,1} - \boldsymbol{e}_{i^*,1}) \neq \boldsymbol{0} \bmod q$. Therefore, based on the above analysis, we conclude that with a probability $\epsilon' \ge \epsilon/(2N) \cdot (1 - (7/9)^\kappa) \cdot (1 - \exp^{-\widetilde{\mathcal{O}}(n)})$, $\hat{\boldsymbol{e}}$ will satisfy $\hat{\boldsymbol{A}} \cdot \hat{\boldsymbol{e}} = \boldsymbol{0} \bmod q$, $0 \neq \|\hat{\boldsymbol{e}}\|_\infty \le 2\beta \cdot (1 + \omega(\sqrt{\log m})) = \mathrm{poly}(m)$.