Forward-Secure Revocable Secret Handshakes from Lattices

Zhiyuan An[1,2], Jing Pan[3], Yamin Wen[2,4], and Fangguo Zhang[1,2](✉)

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
anzhy@mail2.sysu.edu.cn, isszhfg@mail.sysu.edu.cn

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

[3] State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China
jinglap@aliyun.com

[4] School of Statistics and Mathematics, Guangdong University of Finance and Economics, Guangzhou 510320, China
wenyamin@gdufe.edu.cn

Abstract. Secret handshake (SH), as a fundamental privacy-preserving primitive, allows members from the same organization to anonymously authenticate each other. Since its proposal by Balfanz et al., numerous constructions have been proposed, among which only the ones separately designed by Zhang et al. over coding and An et al. over lattice are secure against quantum attacks. However, none of known schemes consider the issue of key exposure, which is a common threat to cryptosystem implementations. To guarantee users' privacy against the key exposure attack, forward-secure mechanism is believed to be a promising countermeasure, where secret keys are periodically evolved in such a one-way manner that, past transactions of users are protected even if a break-in happens.

In this work we formalize the model of forward-secure secret handshake and present the first lattice-based instantiation, where ABB HIBE is applied to handle key evolution process through regarding time periods as hierarchies. In particular, dynamic revocability is captured by upgrading the static verifier-local revocation techniques into updatable ones. To achieve anonymous handshake with ease, we present a generic way of transforming zero-knowledge argument systems termed as Fiat-Shamir with abort, into mutual authentication protocols. Our scheme is proved secure under the Short Integer Solution (SIS) and Learning With Errors (LWE) assumptions in the random oracle model.

Keywords: Secret handshake · Lattice cryptography · Forward security · User revocation · Zero-knowledge

1 Introduction

Secret Handshake, introduced by Balfanz et al. [7], is a fundamental anonymity primitive, where potential members form different groups and conduct an interactive protocol to authenticate each other. The mutual handshake is successful if and only if both parties belong to the same organization. Except for the affiliations, no extra information (including the identities) about the involved members will be leaked. Therefore, secret handshakes provide all-sided privacy-preserving property for enrolled members. To date, many practical applications of secret handshakes in social networks have been explored, such as online dating, mobile access [30] and e-healthcare [39], etc.

Unfortunately, most online infrastructures offering authentication interface run in the unprotected environment, where key exposure can be one of the most fatal damages as it thoroughly destroys the expected security [36]. Forward-secure mechanism [8,21], is a promising method to address the above problem, which preserves the validity of users' past actions. Its core design is a key evolving technique that proceeds as follows. The lifetime of the related scheme is divided into discrete periods. Upon each new period advancing, a subsequent secret key is evolved from the current one via a one-way key update algorithm. Then the current key is erased from the user's records. Due to the one-wayness of the evolving method, the security of past periods' keys is preserved after a break-in at some group member. By leveraging this technique, numerous cryptographic primitives supporting forward security have been constructed, such as digital signature [1,12,32] and public-key encryption [10,15].

Compared with the cases of ordinary signatures or authentication protocols, key exposure can be more damaging to SH systems. Once an adversary obtains the exposed credential of some legitimate user, it can impersonate that user to authenticate any others from the same group, such that a successful handshake no longer ensures a valid authentication. Besides, due to the anonymity of interactions, key exposure essentially undermines the whole group as it invalidates all previously completed handshakes within that group, regardless of who the participants were. Moreover, malicious users, who communicated with honest ones (after authenticating each other) and got their handshakes opened, may defend themselves by giving away their credentials over the Internet and claiming that some hacker conducted the behaviors. Albeit the potential threats of credentials being compromised, no previous SH schemes considered this issue, except the construction of Wen et al. [38], where users are provided with a series of random credentials corresponding to discrete time periods. However, this countermeasure obviously brings huge storage cost and falls short of being succinct. On the other side, to conceptually explore the security against key exposure in SH, it would be better to first formalize the related generic model.

Our Contributions. This work exploits the field of forward-secure secret handshakes. Our contributions are summarized in the following.

- By carefully reforming the desired functionalities and security notations, we adapt the basic model of SH to the forward-secure setting.
- Under the above model, we present a lattice-based SH scheme. In particular,
  - We upgrade the static verifier-local revocation method into time-advanced updatable one, and prove that the iterative process works exactly in a zero-knowledge manner.

– We show how to transform a special type of zero-knowledge system into an anonymous mutual authentication protocol, in a generic manner.

Other Related Works. Following Balfanz et al.'s pioneering work [7], early SH constructions [17,22,43] employed one-time pseudonyms, which bears huge storage cost. One more efficient method is to apply reusable credentials. Xu and Yung [40] first designed a such scheme with weaker unlinkability. Ateniese et al. [6] proposed an efficient unlinkable secret handshake scheme in the standard model. Subsequently, Jarecki and Liu [23] proposed a framework for unlinkable secret handshake scheme that supports both traceability and revocation. From then on, various SH schemes offering different functionalities were proposed [20,37,39]. However, these schemes are designed over number-theoretic assumptions and are vulnerable to quantum attacks. As all we know, only the ones separately proposed by Zhang et al. over coding theory [42] and An et al. [5] over lattice are secure against quantum computations.

Note that none of existing SH schemes has formally considered the issue of key exposure, let alone propose available schemes over post-quantum candidates.

Organization. In Sect. 2, we recall some necessary background and techniques. Model and security requirements of forward-secure secret handshakes are provided in Sect. 3. Section 4 describes the supporting zero-knowledge argument system, which is further modified to support mutual authentication in a handshake. In Sect. 5, we present our lattice-based secret handshake scheme, followed by the analysis of efficiency and security.

2 Preliminaries

Vectors will be denoted in bold lower-case letters and matrices will be denoted in bold upper-case letters. Let $\|\cdot\|$ and $\|\cdot\|_\infty$ denote the Euclidean norm (ℓ_2) and infinity norm (ℓ_∞), respectively. The Euclidean norm of matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$ with columns $(\mathbf{b}_i)_{i\le n}$ is denoted by $\|\mathbf{B}\| = \max_{i\le n}\|\mathbf{b}_i\|$. If $\mathbf{B}$ is full column-rank, let $\widetilde{\mathbf{B}}$ denote its Gram-Schmidt orthogonalization. The concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times k}$ is denoted by $[\mathbf{A}|\mathbf{B}]$. For positive integer n, let $[n]$ denote the set $\{1,\ldots,n\}$. If S is a finite set, denote by $U(S)$ the uniform distribution over S and by $x \hookleftarrow D$ sampling x according to the distribution D.

2.1 Background on Lattices

Classic Lattices and Gaussian Distribution. Let $n, m, q \in \mathbb{Z}^+$ with $q > 2$. For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define two lattices as $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\cdot\mathbf{x} = \mathbf{0} \mod q\}$ and $\Lambda^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\cdot\mathbf{x} = \mathbf{u} \mod q\}$. For a real $\sigma > 0$, a vector $\mathbf{c} \in \mathbb{R}^n$ and n-dimensional lattice L, define the function $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$. The discrete Gaussian distribution over L with parameter σ and center $\mathbf{c}$ is defined as $D_{L,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(L)}$ (write $D_{L,\sigma}(\mathbf{x})$ for short when $\mathbf{c} = 0$).

Lemma 1 **([19,29]).** *Given integers n, $q \geq 2$, and $\sigma \geq \omega(\sqrt{\log n})$. we have* $\Pr_{\mathbf{x} \hookleftarrow D_{\mathbb{Z}^n,\sigma}}[\|\mathbf{x}\|_\infty \geq \sigma \cdot \log n]$ *is negligible.*

Lattice Algorithms. The following facts describe the algorithms for trapdoor generation, Gaussian sampling, lattice basis randomization and delegations.

Lemma 2 **([4]).** *Given integers $n > 0$, $m = O(n \log n)$, $q \geq 2$, this PPT algorithm* $\mathsf{TrapGen}(n, m, q)$ *returns a matrix pair $(\mathbf{A}, \mathbf{T_A})$ satisfies that i) $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is within negligible statistical distance from uniform. ii) $\mathbf{T_A}$ is a basis of $\Lambda^\perp(\mathbf{A})$ and $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.*

Lemma 3 **([19]).** *Given matrices $\mathbf{A} \in \mathbb{Z}^{n\times m}$, $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ as a basis of $\Lambda^\perp(\mathbf{A})$, vector $\mathbf{u} \in \mathbb{Z}_q^n$ and gaussian parameter $\sigma \geq \omega(\sqrt{\log n})\cdot\|\widetilde{\mathbf{T_A}}\|$, this PPT algorithm* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{u}, \sigma)$ *returns a vector $\mathbf{v} \in \Lambda^{\mathbf{u}}(\mathbf{A})$ sampled from a distribution statistically close to $D_{\Lambda^{\mathbf{u}}(\mathbf{A}),\sigma}$.*

Lemma 4 **([16]).** *Given matrix $\mathbf{T_A}$ be a basis of lattice $\Lambda^\perp(\mathbf{A})$ and gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega\sqrt{\log n}$, this PPT algorithm* $\mathsf{RandBasis}(\mathbf{T_A}, \sigma)$ *outputs a new a basis $\mathbf{T}'_\mathbf{A}$ of $\Lambda^\perp(\mathbf{A})$ such that $\|\mathbf{T}'_\mathbf{A}\| \leq \sigma \cdot \sqrt{m}$ and the distribution of $\mathbf{T}'_\mathbf{A}$ does not depend on $\mathbf{T_A}$ up to a statistical distance.*

Besides, the following depicts that lattice basis can be efficiently delegated and simulated, which will be used in the user key update and security proof.

Lemma 5 **([2,16]).** *Set $\sigma_R = \sqrt{n \log q} \cdot \omega(\sqrt{\log m})$, let $\mathcal{D}_{m\times m}$ denote the distribution of matrices in $\mathbb{Z}^{m\times m}$ defined as $(D_{\mathbb{Z}^m,\sigma_R})^m$ conditioned on the sampled matrix being "$\mathbb{Z}_q$-invertible". Given matrices $\mathbf{A} \in \mathbb{Z}^{n\times m}$, $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ as a basis of $\Lambda^\perp(\mathbf{A})$, $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$ as a product of ℓ matrices sampled from $\mathcal{D}_{m\times m}$. Then:*

1. *Let $\mathbf{A}' \in \mathbb{Z}^{n\times m'}$ be any matrix containing $\mathbf{A}$ as a submatrix. This deterministic polynomial-time algorithm* $\mathsf{ExtBasis}(\mathbf{T_A}, \mathbf{A}')$ *outputs a basis $\mathbf{T}_{\mathbf{A}'}$ of $\Lambda^\perp(\mathbf{A}')$ with $\|\widetilde{\mathbf{T}_{\mathbf{A}'}}\| = \|\widetilde{\mathbf{T_A}}\|$.*
2. *This PPT algorithm* $\mathsf{SampleR}(1^m)$ *outputs a matrix $\mathbf{R} \in \mathbb{Z}^{m\times m}$ from a distribution that is statistically close to $\mathcal{D}_{m\times m}$.*
3. *Let Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\|\cdot(\sigma_R\sqrt{m}\omega(\log^{1/2} m))^\ell\cdot\omega(\log m)$. This PPT algorithm* $\mathsf{BasisDel}(\mathbf{A}, \mathbf{R}, \mathbf{T_A}, \sigma)$ *outputs a basis $\mathbf{T_B}$ of $\Lambda^\perp(\mathbf{A}\mathbf{R}^{-1})$ distributed statistically close to the distribution* $\mathsf{RandBasis}(\mathbf{T}, \sigma)$*, where $\mathbf{T}$ is an arbitrary basis of $\Lambda^\perp(\mathbf{A}\mathbf{R}^{-1})$ satisfying $\|\widetilde{\mathbf{T}}\| < \sigma/\omega(\sqrt{\log m})$.*
4. *This PPT algorithm* $\mathsf{SampleRwithBasis}(\mathbf{A})$ *outputs a matrix $\mathbf{R}$ sampled from a distribution statistically close to $\mathcal{D}_{m\times m}$ and a basis $\mathbf{T_B}$ of $\Lambda^\perp(\mathbf{A}\mathbf{R}^{-1})$ having $\|\widetilde{\mathbf{T_B}}\| \leq \sigma_R/\omega(\sqrt{\log m})$.*

Computational Lattice Problems. We recall the definitions and hardness results of SIS, ISIS and LWE, on which the security of our scheme provably relies.

Definition 1 ([3,19]). *Given parameters m, q, β the functions of n, uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$ and matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathsf{SIS}_{n,m,q,\beta}$ (resp., $\mathsf{ISIS}_{n,m,q,\beta}$) demands to find a non-zero vector $\mathbf{x} \in \Lambda^{\perp}(\mathbf{A})$ (resp., $\Lambda^{\mathbf{u}}(\mathbf{A})$) such that $\|\mathbf{x}\| \le \beta$.*

For any $q \geq \beta \cdot \omega(\sqrt{n \log n})$, hardness of $\mathsf{SIS}_{n,m,q,\beta}$ and $\mathsf{ISIS}_{n,m,q,\beta}$ is given by a worst-case to average-case reduction from SIVP_γ for some $\gamma = \beta \cdot \widetilde{\mathcal{O}}(\sqrt{nm})$.

Definition 2 ([35]). *Let $n, m \geq 1, q \geq 2$, and Let χ be a probability distribution over $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \hookleftarrow \chi$, and outputting the pair $(\mathbf{a}, \mathbf{a}^\top \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. Decision-$\mathsf{LWE}_{n,q,\chi}$ problem is to distinguish m samples from $A_{\mathbf{s},\chi}$ (let $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$) and m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Search-$\mathsf{LWE}_{n,q,\chi}$ problem is to find the uniformly random $\mathbf{s}$ given m samples from $A_{\mathbf{s},\chi}$.*

For prime power q, $\beta \geq \sqrt{n}\mathcal{O}(\log n), \gamma = \widetilde{\mathcal{O}}(nq/\beta)$, and a β-bounded distribution χ, *decision*-$\mathsf{LWE}_{n,q,\chi}$ problem is as least as hard as SIVP_γ. Also, *decision*-LWE is proved to be equivalent to *search*-LWE up to some polynomial increase of the sample number m (see [33]). In this work, for a discrete Gaussian distribution χ (i.e., $\chi = D_{\mathbb{Z}_m,\sigma}$), we write *decision*-$\mathsf{LWE}_{n,q,\chi}$ as $\mathsf{LWE}_{n,q,\sigma}$ for short.

2.2 Efficient Signature Scheme from Lattices

Libert *et al.* in [24] proposed a signature scheme (extended from the Böhl *et al.*'s signature [9]) with efficient protocols, of which a variant will serve for the joining phase in our scheme. The scheme utilizes the following parameters: security parameter λ; integers $\ell = \mathsf{poly}(\lambda)$, $n = \mathcal{O}(\lambda)$, $q = \widetilde{\mathcal{O}}(n^4)$ and $m = 2n\lceil \log q \rceil$; Gaussian parameter $\sigma = \Omega(\sqrt{n \log q})$; public key $\mathsf{pk} := (\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{u})$ and private key $\mathsf{sk} := \mathbf{T_G}$, where $(\mathbf{G}, \mathbf{T_G}) \leftarrow \mathsf{TrapGen}(n, m, q)$, $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n\times m/2})$, $\mathbf{G}_i, \mathbf{D}_i \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ for $i \in \{0, 1\}$ and $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

To make a signature on $\mathbf{m} \in \{0, 1\}^m$, one first chooses $i \hookleftarrow [2^\ell]$ and builds the encoding matrix $\mathbf{G}_i = [\mathbf{G}|\mathbf{G}_0 + i\mathbf{G}_1]$ with its delegated basis $\mathbf{T}_i$, then computes the chameleon hash of $\mathbf{m}$ as $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathbf{m}$ with vector $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_M)$. The resulted signature is $sig = (i, \mathbf{d}, \mathbf{r})$ where $\mathbf{d} \in \mathbb{Z}_q^{2m}$ is a short vector in $D_{\Lambda^{\mathbf{u}_M}(\mathbf{G}_i),\sigma}$. The verification step is conducted via checking if $\mathbf{G}_i \cdot \mathbf{d} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathbf{m})$, $\|\mathbf{d}\| < \sigma\sqrt{2m}$ and $\|\mathbf{r}\| < \sigma\sqrt{m}$. It was proved in [24] that the signature above is secure under chosen-message attacks under the SIS assumption.

2.3 Zero-Knowledge Argument Systems

In a zero-knowledge argument of knowledge (ZKAoK) system [13], a prover proves the possession of some witness for an NP relation to a verifier, without revealing any additional information. Generally, a secure ZKAoK must satisfy 3 requirements: *completeness, proof of knowledge* and *(honest verifier) zero knowledge.*

Yang et al. [41] have proposed an efficient lattice-based ZKAoK for relation:

$$\mathcal{R} = \{(\boldsymbol{M}, \boldsymbol{y}, \mathcal{L}), (\boldsymbol{x}) : \boldsymbol{M} \cdot \boldsymbol{x} = \boldsymbol{y} \wedge \forall (i, j, k) \in \mathcal{L}, \boldsymbol{x}[i] = \boldsymbol{x}[j] \cdot \boldsymbol{x}[k]\},$$

where $\boldsymbol{x} \in \mathbb{Z}_q^{\mathfrak{n}}$ is the secret witness and set $\mathcal{L}$ defines quadratic constraints over $\boldsymbol{x}$. The protocol can be further transformed into an $\mathsf{NIZKAoK}$ consisting of two algorithms `Prove` and `Verify` via Fiat-Shamir heuristic. `Prove` produces commitments $cmt = (cmt_s, cmt_r)$, a challenge $ch = \mathcal{G}(cmt, \cdot)$ where $\mathcal{G}$ is a random oracle, and some responses rsp related to ch and cmt. Then (cmt_s, ch, rsp) is sent to a verifier, on input which `Verify` recovers reserved commitment cmt_r and computes ch' by assembling cmt, it finally checks $ch' \overset{?}{=} ch$ to verify the ZK proof. In this paper, we will adapt the $\mathsf{NIZKAoK}$ to an anonymous mutual authentication protocol.

Theorem 1. *The scheme described in Fig. 2 of [41] is a secure* $\mathsf{NIZKAoK}$ *with negligible completeness and soundness error, under the hardness assumptions of* SIS *and* LWE*, and has well-designed simulator and knowledge extractor.*

2.4 LWE-Based Key Exchange

Derived from the design in [18,34] describes an LWE-based key exchange using reconciliation mechanism, which yields keys indistinguishable from random. Let χ be a probability distribution over $\mathbb{Z}_q$, integer θ be the number of bits for key extraction and $\mathbf{K}$ is a public matrix. The following protocol is utilized to produce a communication key in our scheme.

- *Alice* samples a secret matrix $\mathbf{S}_a \hookleftarrow \chi(\mathbb{Z}_q^{n\times m})$ and a small noise $\mathbf{E}_a \hookleftarrow \chi(\mathbb{Z}_q^{n\times m})$. Then, she computes $\mathbf{C}_a = \mathbf{K} \cdot \mathbf{S}_a + \mathbf{E}_a$ and sends it to *Bob*.
- Receiving $\mathbf{C}_a$, *Bob* chooses his secret matrix $\mathbf{S}_b \hookleftarrow \chi(\mathbb{Z}_q^{m\times n})$ and computes $\mathbf{C}_b = \mathbf{K} \cdot \mathbf{S}_b + \mathbf{E}_b$, where $\mathbf{E}_b \hookleftarrow \chi(\mathbb{Z}_q^{m\times n})$. Then he samples a noise $\mathbf{E}'_b \hookleftarrow \chi(\mathbb{Z}_q^{m\times m})$ and sets $\mathbf{V}_b = \mathbf{S}_b \cdot \mathbf{C}_a + \mathbf{E}'_b$. Such that he extracts the shared secret key $\mathbf{K}_b = \mathsf{Extract}(\mathbf{V}_b)$, namely, $\mathbf{K}_b[i, j] = \mathsf{round}((2^\theta/q_1) \cdot \mathbf{V}_b[i, j]) \mod 2^\theta$.. *Bob* also produces a check matrix $\mathbf{M} = \mathsf{Check}(\mathbf{V}_b)$ as $\mathbf{M}[i, j] = \mathsf{floor}((2^{\theta+1}/q_1) \cdot \mathbf{V}_b[i, j]) \mod 2$. Finally, *Bob* sends $(\mathbf{C}_b, \mathbf{M})$ to Alice.
- With $(\mathbf{C}_b, \mathbf{M})$, *Alice* computes $\mathbf{V}_a = \mathbf{C}_b \cdot \mathbf{S}_a$ and obtains $\mathbf{K}_a = \mathsf{Recon}(\mathbf{V}_a, \mathbf{M})$ via $\mathbf{K}_a[i, j] = \mathsf{round}((2^\theta/q_1) \cdot \mathbf{V}_a[i, j] + \frac{1}{4} \cdot (2\mathbf{M}[i, j] - 1)) \mod 2^\theta$.

Theorem 2. **([18]).** *The key exchange above produces the same shared key, i.e.,* $\mathbf{K}_a = \mathbf{K}_b$*, with overwhelming probability via applying suitable parameters.*

3 Model of Forward-Secure Secret Handshakes

As its analogues of group signature [26,32], we consider SH schemes having lifetime divided into discrete time periods, at the beginning of which group members autonomously update the secret keys for forward security. Let time period $t = 0$ be the moment that an SH system is activated, and assume that a handshake always finishes during the period at which it starts. The syntax of forward secure secret handshake (FSSH) is formalized as follows.

- Setup. Given a security parameter $\lambda \in \mathbb{N}$, this algorithm, possibly run by a trusted party or decentralized setting, generates public parameter par.
- $\mathsf{CreateGroup}$. Given par, group authority (GA) invokes this algorithm to create a group. It publishes the group public key gpk and retains secret key gsk.
- $\mathsf{AddMember}$. This protocol is run between a potential user U and GA to enroll the user into a chosen group. At time period t, U generates individual key pair $(\mathsf{upk}, \mathsf{usk}_t)$ and sends upk to GA. If it terminates successfully, GA issues a group credential cred_0 (including a unique group identity ID) to U and adds cred_0 to user's registration table Reg.
- $\mathsf{Update}_\mathsf{U}$. On input user's private pair $(\mathsf{cred}_{t-1}, \mathsf{usk}_{t-1})$ at the beginning of time period t, this one-way algorithm evolves it into $(\mathsf{cred}_t, \mathsf{usk}_t)$.
- $\mathsf{Handshake}$. This is a mutual authentication protocol between two participants (A, B). It outputs 1 and produces a session key for both active parties at the current time period t if and only if they belong to the same group.
- $\mathsf{TraceMember}$. Given a handshake transcript, GA runs this algorithm to trace the involved users, or outputs $\perp$ to indicate a failure.
- $\mathsf{RemoveMember}$. This algorithm is invoked by GA to revoke an active member. GA also publishes some updated group information of that group for current time period, such that users can conduct revocation check.

Based on the considerations in [7,26], we reform the security requirements an FSSH must satisfy as *Completeness*, *Forward impersonator resistance*, *Detector resistance* and *Backward unlinkability*, all of which are defined via the corresponding experiments. Hereafter we use CU and CG to denote the corruption list of users and groups, respectively.

Completeness demands that $\mathsf{Handshake}$ outputs 1 with overwhelming probability if both participants are active with updated secret keys and belong to the same group. Moreover, $\mathsf{TraceMember}$ can always identify the involved users. For plain description, we define an auxiliary polynomial-time algorithm $\mathsf{IsActive}(\mathsf{ID}, t)$: outputs 1 if ID is active at current time period t and 0 otherwise.

Definition 3. *The completeness is achieved if the following experiment returns 1 with negligible probability.*

Experiment: $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{COM}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, $(\mathsf{gpk}, \mathsf{gsk}) \leftarrow \mathsf{CreateGroup}(\mathsf{par})$.
$\{(\mathsf{ID}_0, \mathsf{cred}_{0\|\bar{t}}, \mathsf{usk}_{0\|\bar{t}}), (\mathsf{ID}_1, \mathsf{cred}_{1\|\bar{t}}, \mathsf{usk}_{1\|\bar{t}})\} \leftarrow \mathsf{AddMember}(\mathsf{gpk}, \mathsf{gsk}, \bar{t})$
$\mathsf{IsActive}(\mathsf{ID}_b, t) = 1 \wedge \mathsf{usk}_{b\|t} \leftarrow \mathsf{Update}_\mathsf{U}(\mathsf{cred}_b, \mathsf{usk}_{b\|t-1})$ for $b \in \{0, 1\}$.
If $\mathsf{Handshake}(\mathsf{ID}_0, \mathsf{ID}_1, t) = 0$ with transcript T or $\mathsf{TraceMember}(T) \notin \{\mathsf{ID}_0, \mathsf{ID}_1\}$,
then return 1 else retun 0.

Forward impersonator resistance requires that it is infeasible for any PPT adversary $\mathcal{A}$ to impersonate an uncorrupted user, or some corrupted user at the period preceding the one where she was broken into, even if it can corrupt all the users and groups (except the chosen group) via accessing the following oracles.

Below are oracles that entitle $\mathcal{A}$ to obtain exterior information of an FSSH.

- $\mathsf{KeyP}(\mathsf{par})$ simulates to create a new group and returns gpk to $\mathcal{A}$.
- $\mathsf{HS}(U, V)$ simulates a two-party handshake by generating the transcripts during the interaction.
- $\mathsf{Trace}(T)$ returns the participant of transcript T. Hereafter we require that T is not generated from the challenging oracles.
- $\mathsf{Remove}(U)$ simulates to revoke user U from her G, it also updates the corresponding group information at current period t.

The other oracles below enable $\mathcal{A}$ to break into the internal of an FSSH.

- $\mathsf{CorU}(U, G)$ is a user corruption oracle. It returns user's cred and usk of U in group G to $\mathcal{A}$ at period t, then it adds (ID_U, G, t) to CU.
- $\mathsf{AddU}(U, G)$ enrolls a user U whose key pair is chosen by $\mathcal{A}$ to G at period t. It also adds (ID_U, G, t) to CU. Compared with CorU, AddU endows $\mathcal{A}$ with more power to create dummy users or perform injection attacks.
- $\mathsf{KeyG}(\mathsf{par})$ returns msk of some group G to $\mathcal{A}$ and adds G to CG, meaning that G is under the control of $\mathcal{A}$.

Now we describe the challenge game of *forward impersonator resistance.*

- $\mathsf{Chal}^{\mathsf{F\text{-}IR}}(\mathsf{ID}, G, t)$ simulates ID of group G and executes a handshake with $\mathcal{A}$ using the updated secret key of ID at period t. It returns 1 if the protocol outputs 1 and 0 otherwise.

Hereafter we denote the transcript of $\mathcal{A}$ during the challenge game as T.

Definition 4. *Forward impersonator resistance is achieved if, for any $\mathcal{A}$, the following experiment returns 1 with negligible probability.*
Experiment*:* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{F\text{-}IR}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, CG, $\mathsf{CU} := \emptyset$.
$(\mathsf{ID}^*, G^*, t^*) \leftarrow \mathcal{A}^{\mathsf{KeyP},\mathsf{HS},\mathsf{Trace},\mathsf{Remove},\mathsf{CorU},\mathsf{AddU},\mathsf{KeyG}(\neg G^*)}(\mathsf{par})$.
If $\mathsf{Chal}^{F\text{-}IR}(\mathsf{ID}^*, G^*, t^*) = 0$, return 0. Else if for $\mathsf{ID}' \leftarrow \mathsf{TraceMember}(T)$: $(\mathsf{ID}', \cdot, \cdot) \notin \mathsf{CU}$ or $t^* < t$ for $(\mathsf{ID}', \cdot, t) \in \mathsf{CU}$, return 1. Else return 0.

Detector resistance makes sure that $\mathcal{A}$ cannot succeed when he activates a handshake with an honest and active user to identify her affiliation at the chosen time period, even if it can corrupt all the users and groups (except the chosen group). The related challenge game is described as follows.

- $\mathsf{Chal}_b^{\mathsf{DR}}(\mathsf{ID}, G, t)$ chooses a random bit $b \in \{0, 1\}$. For $b = 0$, it simulates ID from G to handshake with $\mathcal{A}$. For $b = 1$, it simulates an arbitrary (active) user ID_r to handshake with $\mathcal{A}$. Then $\mathcal{A}$ guesses the value of b as b^*.

Definition 5. *Detector resistance is achieved if, for any $\mathcal{A}$, the absolute difference of probability of outputting 1 between* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR-1}}$ *and* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR-0}}$ *is negligible.*

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

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, CG, $\mathsf{CU} := \emptyset$.
$(\mathsf{ID}^*, G^*, t^*) \leftarrow \mathcal{A}^{\mathsf{KeyP,HS,Trace,Remove,CorU,AddU,KeyG}(\neg G^*)}(\mathsf{par})$, $\mathsf{Chal}_b^{\mathsf{DR}}(\mathsf{ID}^*, G^*, t^*)$,
holding that for $\mathsf{ID}' \leftarrow \mathsf{TraceMember}(T)$, $(\mathsf{ID}', \cdot, \cdot\,) \notin \mathsf{CU}$ or $\mathsf{IsActive}(\mathsf{ID}', t^*) = 0$.
Return $b^* \leftarrow \mathcal{A}^{\mathsf{KeyP,HS,Trace,Remove,CorU,AddU,KeyG}(\neg G^*)}(\mathsf{par})$.

Backward Unlinkability ensures that no adversary can distinguish whether two handshakes (executed during two distinct periods) involve the same honest user, even if it can corrupt any user and any group (except the chosen pair), and that user is later revoked. Below is the related challenge game.

- $\mathsf{Chal}_b^{\mathsf{B\text{-}Unlink}}(\mathsf{ID}_0, G_0, \mathsf{ID}_1, G_1, t)$ first picks a random bit $b \in \{0,1\}$, then it successionally simulates ID_0 and ID_b to handshake with $\mathcal{A}$ using evolved secret. Finally $\mathcal{A}$ guesses the value of b as b^*.

Definition 6. *Backward unlinkability is achieved if, for any $\mathcal{A}$, the absolute difference of probability of outputting 1 between $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{B\text{-}Unlink-1}}$ and $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{B\text{-}Unlink-0}}$ is negligible.*
Experiment: $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{B\text{-}Unlink-b}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, CoG, $\mathsf{CoU} := \emptyset$.
$(\mathsf{ID}_0, G_0, \mathsf{ID}_1, G_1, t) \leftarrow \mathcal{A}^{\mathsf{KeyP,HS,Trace,Remove,CorU,AddU,KeyG}(\neg G^*)}(\mathsf{par})$,
holding that $G_i \notin \mathsf{CG} \wedge (\mathsf{ID}_i, G_i, \cdot\,) \notin \mathsf{CU}$ for $i \in \{0,1\}$.
$b^* \leftarrow \mathcal{A}^{\mathsf{Chal}_b^{B\text{-}Unlink}(\mathsf{ID}_0, G_0, \mathsf{ID}_1, G_1, t)}(\mathsf{par})$. Return b^*.

Note that if $\mathcal{A}$ has corrupted some user of G_i for $i \in \{0,1\}$, then he is only allowed to choose target users within that group, i.e., $G_0 = G_1$.

4 The Supporting Zero-Knowledge Layer

In this section, we first construct a system that allows obtaining ZKAoK for some relations, which are linear equations within users' credential and secret key of our FSSH scheme. Then we clarify why ZK argument cannot be directly used in a handshake procedure, for this reason we further present a generic way of transforming ZK systems termed as Fiat-Shamir with abort into mutual authentication protocols, where participants can "handshake" with each other and negotiate a session key.

Below we extensively use the decomposition techniques in [24,27]. Namely, for any integer $\beta > 0$, let $\delta_\beta = \lceil \log(\beta+1) \rceil$ and $\beta_j = \lfloor \frac{\beta + 2^{j-1}}{2^j} \rfloor$ $\forall j \in [1, \delta_\beta]$. Then any $i \in [0, \beta]$ can be decomposed as $i = \mathbf{h}_\beta \cdot \mathsf{idec}_\beta(i)$, where $\mathbf{h}_\beta = (\beta_1, \ldots, \beta_{\delta_\beta})$ and idec_β is a binary function. Further, [25] build two more functions for decomposing vectors and matrices: $\mathsf{vdec}_{m,\beta} : [0,\beta]^m \rightarrow \{0,1\}^{m\delta_\beta}$; $\mathsf{mdec}_{n,m,q} : \mathbb{Z}_q^{m \times n} \rightarrow \{0,1\}^{nm\delta_{q-1}}$. (see [25] or the full version of this paper for detailed definitions.)

4.1 ZKAoK System for Proving a Valid User

Now we describe the system that produces ZK arguments for users' secret. Given the same situation as that of Handshake in Sect. 5 with extra setting: $\mathbf{H}_{m,\beta} = \mathbf{I}_m \otimes \mathbf{h}_\beta$, $t_i = t_{add} + i$ for $i \in [t^*]$, $\mathbf{h}_N = (N_1, \ldots, N_\ell)$, $\mathbf{a}' = (\mathbf{a}_1'^\top \ \ldots \ \mathbf{a}_n'^\top)^\top = \mathsf{mdec}_{n,m,q}(\mathbf{A}^\top)$, $\mathbf{b} = \mathsf{vdec}_{2n,q-1}(\mathbf{h})$, $\mathbf{w} = (\mathbf{w}_1^\top \ \ldots \ \mathbf{w}_n^\top)^\top = \mathsf{mdec}_{n,m,q}(\mathbf{A}_t^\top)$ and $\mathbf{z} = \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathbf{b})$, the desired system is summarized as follows.

Public Input: Matrices $\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{B}, \mathbf{P}, \mathbf{W}$; Vectors $\mathbf{u}, \mathbf{t}, \mathbf{w}, \mathbf{k}$; Integer t^*. System public parameter par.

Prover's Witness: Vectors and Matrices which satisfy the following constraints

$$\begin{cases} \mathsf{ID} = \mathbf{i} \in \{0,1\}^\ell, \mathsf{urt}_t = \mathbf{q} \in \mathbb{Z}_q^n, \mathbf{d} = (\mathbf{d}_1^\top \ \mathbf{d}_2^\top)^\top \in \{-\beta, \beta\}^{2m}, \\ \mathbf{r} \in \{-\beta, \beta\}^m, \mathbf{a}' \in \{0,1\}^{nmk}, \mathbf{b} \in \{0,1\}^{2nk}, \mathbf{v} \in \{-\beta_d, \beta_d\}^m \\ \mathbf{e} \in \{-B, B\}^m, \mathbf{s} \in \{-B, B\}^n, \mathbf{e}_1 \in \{-B, B\}^m, \mathbf{e}_2 \in \{-B, B\}^\ell, \\ \mathbf{A} = [\mathbf{a}_1 | \ldots | \mathbf{a}_n] \in \mathbb{Z}_q^{n\times m}, \mathbf{A}_t^\top = [\mathbf{a}_{t,1} | \ldots | \mathbf{a}_{t,n}] \in \mathbb{Z}_q^{m \times n}. \end{cases} \tag{1}$$

Prover's Goal: Convince the verifier in zero-knowledge that the following set of modular linear equations holds[1] (under the same modulus q):

$$\begin{cases} \mathbf{F} \cdot \mathbf{a} - \mathbf{H}_{2n,q-1} \cdot \mathbf{b} = \mathbf{0}, \\ \mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathbf{b} - \mathbf{H}_{n,q-1} \cdot \mathbf{z} = \mathbf{0}, \\ [\mathbf{G} | \mathbf{G}_0 | N_1 \mathbf{G}_1 | \ldots | N_\ell \mathbf{G}_1] \cdot (\mathbf{d}_1, \mathbf{d}_2, \mathbf{i}[1]\mathbf{d}_2, \ldots, \mathbf{i}[\ell]\mathbf{d}_2)^\top - \mathbf{D} \cdot \mathbf{z} = \mathbf{u}, \\ \mathbf{A} \ (\mathbf{R}_1^{\mathbf{t}[1]})^{-1} \ (\mathbf{R}_2^{\mathbf{t}[2]})^{-1} \ldots (\mathbf{R}_d^{\mathbf{t}[d]})^{-1} - \mathbf{A}_t = \mathbf{0}, \\ \mathbf{A}_t \cdot \mathbf{v} = \mathbf{u}, \\ \mathbf{a}_1 - \mathbf{H}_{n,q-1} \cdot \mathbf{q}' = \mathbf{0}, \\ \mathbf{Q}_1 \cdot \mathbf{q}' + \mathbf{Q}_2 \cdot \mathbf{t}_{add} - \mathbf{q}_0 = \mathbf{0}, \quad \mathbf{q}_{t^*} = \mathbf{q}, \\ \forall i \in [t^*] : (\mathbf{q}_{i-1} - \mathbf{H}_{n,q-1} \cdot \mathbf{q}'_{i-1}, \mathbf{Q}_1 \cdot \mathbf{q}'_{i-1} + \mathbf{Q}_2 \cdot \mathbf{t}_i - \mathbf{q}_i) = (\mathbf{0}, \mathbf{0}), \\ \mathbf{W} \cdot \mathbf{q} + \mathbf{e} = \mathbf{w}, \\ \mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}_1 = \mathbf{c}_1, \\ \mathbf{P}^\top \cdot \mathbf{s} + \mathbf{e}_2 + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{i} = \mathbf{c}_2. \end{cases} \tag{2}$$

Since Set 2 is somewhat complicated, we first design two sub-systems: Π_1 arguing that user's credential is issued via making a signature on her public key, and her secret key is updated rightly with time advances.; Π_2 evidencing that 1) her updatable revocation token is rightly derived from the public key and embedded in an LWE function; 2) her identity is correctly encrypted with ciphertexts $(\mathbf{c}_1, \mathbf{c}_2)$. Then we establish Π_{hs} by combining Π_1 and Π_2.

Build System Π_1. This system covers the first five equations of Set 2. Our goal is to integrate these linear equations into a uniform relation

$$\mathcal{R}_1 = \{(\mathbf{M}_1, \mathbf{y}_1, \mathcal{L}_1), (\mathbf{x}_1) : \mathbf{M}_1 \cdot \mathbf{x}_1 = \mathbf{y}_1 \wedge \mathbf{x}_1 \in \mathsf{cons}_1\}.$$

Let $\boldsymbol{\beta} = (\beta \ \ldots \ \beta)^\top, \boldsymbol{\beta}_d = (\beta_d \ \ldots \ \beta_d)^\top \in \mathbb{Z}_q^m$. First perform the following steps.

[1] We refer readers to Sect. 5 for more information of these equations.

1. Set $\mathbf{r}' = \mathbf{r} + \boldsymbol{\beta} \in [0, 2\beta]^m$, $\mathbf{d}_j'^{\top} = \mathbf{d}_j + \boldsymbol{\beta} \in [0, 2\beta]^m$ for each $j \in \{1, 2\}$ and $\mathbf{v}' = \mathbf{v} + \boldsymbol{\beta}_d \in [0, 2\beta_d]^m$. Decompose $\mathbf{r}', \mathbf{d}_j', \mathbf{v}'$ such that $\mathbf{r}' = \mathbf{H}_{m,2\beta} \cdot \mathbf{r}''$, $\mathbf{d}_j' = \mathbf{H}_{m,2\beta} \cdot \mathbf{d}_j''$ for $j \in \{1, 2\}$ and $\mathbf{v}' = \mathbf{H}_{m,2\beta_d} \cdot \mathbf{v}''$, respectively.
2. Set matrices $\mathbf{G}' = \mathbf{G} \cdot \mathbf{H}_{m,2\beta}$, $\mathbf{G}_0' = \mathbf{G}_0 \cdot \mathbf{H}_{m,2\beta}$, $\mathbf{D}_0' = \mathbf{D}_0 \cdot \mathbf{H}_{m,2\beta}$ and $\mathbf{G}_j' = N_j \mathbf{G}_1 \cdot \mathbf{H}_{m,2\beta}$ for each $j \in [\ell]$. Assemble auxiliary matrices $\mathbf{G}_j'' = -N_j \mathbf{G}_1 \cdot \boldsymbol{\beta}$ for each $j \in [\ell]$, and vectors $\mathbf{u}' = \mathbf{u} + (\mathbf{G} + \mathbf{G}_0) \cdot \boldsymbol{\beta}$, $\mathbf{u}_1 = \mathbf{D} \cdot \boldsymbol{\beta}$.
3. Denote $[\mathbf{G}'|\mathbf{G}_0'|\mathbf{G}_1'|\ldots|\mathbf{G}_\ell']$ and $[\mathbf{G}_1''|\ldots|\mathbf{G}_\ell'']$ as $\bar{\mathbf{G}}'$ and $\bar{\mathbf{G}}''$, respectively.
4. Denote transpose of product $(\mathbf{R}_1^{\mathbf{t}[d]})^{-1} \ldots (\mathbf{R}_d^{\mathbf{t}[1]})^{-1}$ as $\mathbf{R}^{(\mathbf{t})}$. Define $\mathbf{N} = \mathbf{R}^{(\mathbf{t})} \cdot \mathbf{H}_{m,q-1}$, and build the extension matrix $\mathbf{L}_1 = \mathbf{I}_m \otimes \mathbf{N}$, $\mathbf{L}_2 = \mathbf{I}_n \otimes \mathbf{1}^m$.
5. Let $\mathbf{c} = (\mathbf{a}_{t,1}[1]\mathbf{v}[1] \ \ldots \ \mathbf{a}_{t,1}[m]\mathbf{v}[m] \ \ldots \ \mathbf{a}_{t,n}[1]\mathbf{v}[1] \ \ldots \ \mathbf{a}_{t,n}[m]\mathbf{v}[m]) \in \mathbf{Z}_q^{mn}$.

Through the above settings, we can change the target part of Set 2 into:

$$\begin{cases} \mathbf{F} \cdot \mathbf{a} - \mathbf{H}_{2n,q-1} \cdot \mathbf{b} = \mathbf{0}, \\ \mathbf{D}_0' \cdot \mathbf{r}'' + \mathbf{D}_1 \cdot \mathbf{b} - \mathbf{H}_{n,q-1} \cdot \mathbf{z} = \mathbf{u}_1, \\ \bar{\mathbf{G}}' \cdot (\mathbf{d}_1'', \mathbf{d}_2'', \mathbf{i}[1]\mathbf{d}_2'', \ldots, \mathbf{i}[\ell]\mathbf{d}_2'')^\top + \bar{\mathbf{G}}'' \cdot \mathbf{i} - \mathbf{D} \cdot \mathbf{z} = \mathbf{u}', \\ \mathbf{L}_1 \cdot [\mathbf{a}_1'|\ldots|\mathbf{a}_n'] - [\mathbf{a}_{t,1}|\ldots|\mathbf{a}_{t,n}] = \mathbf{0}, \\ (\mathbf{a}_{t,1}^\top \mathbf{v} \ \ldots \ \mathbf{a}_{t,n}^\top \mathbf{v})^\top = \mathbf{u}, \\ \mathbf{H}_{m,2\beta_d} \cdot \mathbf{v}'' - \mathbf{v} - \boldsymbol{\beta}_d = \mathbf{0}. \end{cases} \tag{3}$$

After the above preparations, we can obtain the desired variables as follows:

1. Denote $-\mathbf{H}_{n,q-1}$, $-\mathbf{H}_{2n,q-1}$ and $\mathbf{H}_{m,2\beta_d}$ by $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$, respectively. Build the public matrix $\mathbf{M}_1$ and vector $\mathbf{y}_1$ as

$$\begin{pmatrix} \mathbf{F} & \mathbf{H}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_1 & \mathbf{D}_0' & \mathbf{H}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{D} & \bar{\mathbf{G}}'' & \mathbf{G}' & \mathbf{G}_0' & \mathbf{G}_1' & \ldots & \mathbf{G}_\ell' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{I}_{nm} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{L}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_3 & -\mathbf{I}_m \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{u}_1 \\ \mathbf{u}' \\ \mathbf{0} \\ \mathbf{u} \\ \boldsymbol{\beta}_d \end{pmatrix}.$$

2. The private witness $\mathbf{x}$ can be build as

$$(\ \mathbf{a}'^\top \ \mathbf{b}^\top \ \mathbf{r}''^\top \ \mathbf{z}^\top \ \mathbf{i}^\top \ \mathbf{d}_1''^\top \ \mathbf{d}_2''^\top \ \mathbf{i}[1]\mathbf{d}_2''^\top \ldots \mathbf{i}[\ell]\mathbf{d}_2''^\top \ \mathbf{a}_{t,1}^\top \ldots \mathbf{a}_{t,n}^\top \ \mathbf{c}^\top \ \ \mathbf{v}''^\top \ \ \mathbf{v}^\top)^\top.$$

3. Then set $\mathsf{cons}_1 = \mathcal{L}_{1,1} \cup \mathcal{L}_{1,2} \cup \mathcal{L}_{1,3}$ where

$$\begin{cases} \mathcal{L}_{1,1} = \{(i,i,i)\},\ i \in [1, (m+3)nk + \ell + 3m\delta_{2\beta}]; \\ \mathcal{L}_{1,2} = \{((m+3)nk + \ell + (3m+\ell)\delta_{2\beta} + mn + (u-1)m + v, \\ \qquad\qquad (m+3)nk + \ell + (3m+\ell)\delta_{2\beta} + (u-1)m + v, \\ \qquad\qquad (m+3)nk + \ell + (3m+\ell)\delta_{2\beta} + 2mn + 2m\delta_{\beta_d} + v)\}, \\ \qquad\qquad u \in [n], v \in [m]; \\ \mathcal{L}_{1,3} = \{(i,i,i)\},\ i \in [(m+3)nk + \ell + (3m+\ell)\delta_{2\beta} + 2mn + 1, \\ \qquad\qquad\qquad (m+3)nk + \ell + (3m+\ell)\delta_{2\beta} + 2mn + 2m\delta_{\beta_d}], \end{cases}$$

where $\mathcal{L}_{1,1}$ indicates that $\mathbf{a}', \mathbf{b}, \mathbf{r}'', \mathbf{z}, \mathbf{i}, \mathbf{d}_1''$ and $\mathbf{d}_2''$ are all binary vectors, $\mathcal{L}_{1,2}$ ensures that $\mathbf{c}[(u-1)m+v] = \mathbf{a}_{t,u}[v] \cdot \mathbf{v}[v]$ for $(u,v) \in [n] \times [m]$, and $\mathcal{L}_{1,3}$ ensures that $\mathbf{v}''$ is binary.

Build System $\mathit{\Pi}_2$**.** This system also covers the rest part by a unified relation

$$\mathcal{R}_2 = \{(\mathbf{M}_2, \mathbf{y}_2, \mathcal{L}_2), (\mathbf{x}_2) : \mathbf{M}_2 \cdot \mathbf{x}_2 = \mathbf{y}_2 \wedge \mathbf{x}_2 \in \mathsf{cons}_2\},$$

which evidences the correct embedding of user's revocation token and identity. The concrete construction of $\mathit{\Pi}_2$ is much like that of $\mathit{\Pi}_1$ and we also take some preprocessing.

1. Let $\mathbf{b}_1 = (B \ \dots \ B)^\top \in \mathbf{Z}_q^m$, $\mathbf{b}_2 = (B \ \dots \ B)^\top \in \mathbf{Z}_q^n$ and $\mathbf{b}_3 = (B \ \dots \ B)^\top \in \mathbf{Z}_q^\ell$.
2. Set $\mathbf{e}' = \mathbf{e} + \mathbf{b}_1$, $\mathbf{s}' = \mathbf{s} + \mathbf{b}_2$, $\mathbf{e}_1' = \mathbf{e}_1 + \mathbf{b}_1$ and $\mathbf{e}_2' = \mathbf{e}_2 + \mathbf{b}_3$. Decompose them via functions vdec and mdec to get vectors $\mathbf{e}'', \mathbf{s}'', \mathbf{e}_1'', \mathbf{e}_2''$.
3. Compute time-binding vectors $\mathbf{t}_0' = \mathbf{Q}_2 \cdot \mathbf{t}_{add}$ and $\mathbf{t}_i' = \mathbf{Q}_2 \cdot \mathbf{t}_i$ for $i \in [t^*]$, set $\mathbf{t}' = (\mathbf{t}_0'^\top \ \mathbf{t}_1'^\top \ \dots \ \mathbf{t}_{t^*}'^\top)^\top$. Assemble quasi-diagonal matrices $\mathbf{L}_3$ and $\mathbf{L}_4$ as

$$\mathbf{L}_3 = \begin{pmatrix} -\mathbf{H}_1 & & \\ & \ddots & \\ & & -\mathbf{H}_1 \end{pmatrix}, \quad \mathbf{L}_4 = \begin{pmatrix} -\mathbf{Q}_1 & & \\ & \ddots & \\ & & -\mathbf{Q}_1 \end{pmatrix} \in \mathbb{Z}^{(t^*+1)n \times (t^*+1)(nk)}.$$

4. Set $\mathbf{B}' = \mathbf{B}^\top \cdot \mathbf{H}_{n,2B}$, $\mathbf{P}' = \mathbf{P}^\top \cdot \mathbf{H}_{n,2B}$, $\mathbf{I}' = \lfloor \frac{q}{2} \rfloor \cdot \mathbf{I}_\ell$, $\mathbf{w}' = \mathbf{w} + \mathbf{b}_1$, $\mathbf{c}_1' = \mathbf{c}_1 + \mathbf{b}_1 + \mathbf{B}^\top \cdot \mathbf{b}_2$ and $\mathbf{c}_2' = \mathbf{c}_2 + \mathbf{b}_3 + \mathbf{B}^\top \cdot \mathbf{b}_2$.

Use $\mathbf{H}_4$ and $\mathbf{H}_5$ to denote $\mathbf{H}_{m,2B}$ and $\mathbf{H}_{\ell,2B}$, respectively. Then we can construct the target variables as follows:

1. Build the public matrix $\mathbf{M}_2$ and vector $\mathbf{y}_2$ as

$$\begin{pmatrix} [\mathbf{I}_{(t^*+1)n}|\mathbf{0}] & \mathbf{L}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ [\mathbf{0}|\mathbf{I}_{(t^*+1)n}] & \mathbf{L}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ [\mathbf{0} \ \dots \ \mathbf{W}] & \mathbf{0} & \mathbf{H}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} \ \dots \ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}' & \mathbf{H}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} \ \dots \ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}' & \mathbf{0} & \mathbf{H}_5 & \mathbf{I}' \end{pmatrix}, \quad \begin{pmatrix} \mathbf{0} \\ \mathbf{t}' \\ \mathbf{w}' \\ \mathbf{c}_1' \\ \mathbf{c}_2' \end{pmatrix}.$$

2. Set $\mathbf{x}_2 = (\ \mathbf{a}_1^\top \ \ \mathbf{q}_0^\top \ \dots \ \mathbf{q}_{t^*}^\top \ \ \mathbf{q}'^\top \ \ \mathbf{q}_0'^\top \ \dots \ \mathbf{q}_{t^*-1}'^\top \ \ \mathbf{e}''^\top \ \ \mathbf{s}''^\top \ \ \mathbf{e}_1''^\top \ \ \mathbf{e}_2''^\top \ \ \mathbf{i})^\top$, which has length $\mathfrak{n}_2 = (t^*+2)n + (t^*+1)nk + (2m+n+\ell)\delta_{2B} + \ell$.
3. The constriants over $\mathbf{x}_2$ is $\mathsf{cons}_2 = \{(i,i,i)\}$, $i \in [(t^*+2)n+1, (t^*+2)n+(t^*+1)nk+(2m+n+\ell)\delta_{2B}+\ell]$, indicating that $\mathbf{q}'$, $\mathbf{q}_0', \dots, \mathbf{q}_{t^*-1}'$, $\mathbf{e}''$, $\mathbf{s}''$, $\mathbf{e}_1''$, $\mathbf{e}_2''$ and $\mathbf{i}$ are all binary.

Build System $\boldsymbol{\mathit{\Pi}_{hs}}$**.** We obtain the desired system $\mathit{\Pi}_{hs}$ by instantiating the framework in Sect. 2.3 with $\mathit{\Pi}_1$ and $\mathit{\Pi}_2$. Namely, for the final relation

$$\mathcal{R}_{hs} = \{(\mathbf{M}, \mathbf{y}, \mathcal{L}), (\mathbf{x}) : \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \wedge \mathbf{x} \in \mathsf{cons}\},$$

set $\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$, $\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$ and $\mathsf{cons} = \mathsf{cons}_1 \cup \mathsf{cons}'_2 \cup \mathsf{cons}'_3$, where cons'_2 is simply performing right shift in cons_2 by the size of $\mathbf{x}_1$, and $\mathsf{cons}'_3 = (i, j, j)$ ensures that vector $\mathbf{i}$ in two sub-systems is the same one.

4.2 Transformation to Anonymous Mutual Authentication

Although users in our FSSH scheme can invoke Π_{hs} to obtain ZK proof for their group secrets, they cannot directly send the proof in Handshake. The reason is that the other participant receiving that proof can unilaterally verify its validity, so as to verify the legality of the sender without any further interactions, which obviously violates the demand for mutual authentication.

To fill the gap, below we show how to adapt the Fiat-Shamir-type framework [41], by transforming one-side identification into mutual authentication.

- $\texttt{Prove}_{\texttt{hs}}$: On input public parameter and secret witness, produce commitments $cmt = (cmt_s, cmt_r)$ and $ch = \mathcal{G}(cmt, \cdot)$ the same as the original algorithm. Then additionally compute a mixed challenge $\widetilde{ch} = ch \oplus \mathbf{C}$ where $\mathbf{C}$ is interim matrix in a KE, and generate rsp using $(cmt, \widetilde{ch})$. Finally output $\pi = (cmt_s, \widetilde{ch}, rsp)$.
- $\texttt{Verify}_{\texttt{hs}}$: Recover cmt_r via rsp and $\widetilde{ch}$ as $\texttt{Verify}$ does, then get the original challenge $ch' = \mathcal{G}(cmt, \cdot)$ by assembling $cmt = (cmt_s, cmt_r)$, so as to retrieve the hidden message $\mathbf{C} = ch' \oplus \widetilde{ch}$.

Here the key point is that a receiver can no longer check the validity of the proof by checking $ch' \overset{?}{=} ch$, since he only receives $\widetilde{ch}$. On the other side, a KE element can be recovered to negotiate a communication key for both participants, whose hash value is further dispatched to conduct authentication. This strategy can be seen as a generic way of transforming ZK systems termed as Fiat-Shamir with abort into anonymous mutual authentication, for which a concrete instantiation is detailed in Handshake of our main scheme.

5 FSSH with Revocability from Lattices

In this section, by devising updatable VLR method and adaptively applying the building blocks recalled in Sect. 2, we present the first FSSH with revocability from lattice. To clarify the roadmap on how to make all things work, below we first give some key points of our construction.

When enrolling in a group, potential user first samples her initial public/secret key pair $(\mathbf{A}, \mathbf{T})$ via Trapdoor and sends $\mathbf{A}$ to GA, on which GA produces an unforgeable signature [24] as her credential. Since users retain the secret keys, even a malicious GA can not frame a legal user. To enable periodical key updating, we combine the binary-tree representation technique and ABB HIBE [2]. Namely, each node of the tree is assigned a short-norm invertible matrix $\mathbf{R}_i^b$ for $i \in [d]$ and $b \in \{0, 1\}$, and successive periods are associated with leaves of the binary tree in the LTR order. At the joining period $\mathbf{t} = (\mathbf{t}[1], \ldots, \mathbf{t}[d])$, users extract the corresponding key (trapdoor) $\mathbf{T}_t$ at this leaf for $\mathbf{A}(\mathbf{R}_d^{\mathbf{t}[d]} \ldots \mathbf{R}_1^{\mathbf{t}[1]})^{-1}$ by use of $\mathsf{BasisDel}$. Observe that users can generate possible trapdoors of any leaves from the root key $\mathbf{T}$. Thus, one trivial method of key update is to precompute all possible $\mathbf{T}_t$ and then delete the previous one upon new period advancing. However, as noted in [28], this will bring key size undesirable dependency on T. Considering the level structure of a binary-tree, it suffices to only record the keys for sub-set $\mathsf{Evolve}_{(t \to T-1)}$ [12,26], which contains exactly one ancestor of each leaf between $[t, T-1]$ and has size at most $\log T$. Under this setting, users can update usk_t into usk_{t+1} (consisting of trapdoors for elements in $\mathsf{Evolve}_{(t+1 \to T-1)}$), by repeatedly invoking $\mathsf{BasisDel}$ within $\mathsf{Evolve}_{(t \to T-1)}$.

Now we demonstrate how to achieve revocability by applying VLR mechanism [11], where a revocation token urt is issued to a user and will be published when she is revoked. Similar to the case of key exposure, it is worthwhile to protect user's anonymity of previous behaviors even if her token is revealed (known as backward unlinkability [31]). We tackle this problem by subtly devising an updatable VLR algorithm: $\mathsf{urt}_t = \mathbf{Q} \cdot (\mathsf{vdec}(\mathsf{urt}_{t-1}), \mathbf{t})^\top \in \mathbb{Z}_q^n$. Choosing uniformly random $\mathbf{Q}$, this equation is linked to an ISIS instance, so as to achieve one-wayness of updating. Besides, we embed the time tag into the token to enable synchronous revocation check, such that expired tokens cannot be reused. Finally, to bind urt_t to user's secret, we set the initial token as $\mathbf{Q} \cdot (\mathsf{vdec}(\mathbf{a}_0), \mathbf{t})^\top$, where $\mathbf{a}_0$ is the first column of $\mathbf{A}$. When executing a handshake, users need to demonstrate in zero-knowledge the possession of a valid secret. This task is done by reducing the overall linear relations set to system Π_{hs} designed in Sect. 4.1. In particular, to argue the current urt_t and usk_t are correctly derived from the previous ones and are compatible with each other, we unify a time-advancing chain of iterative equations into a universal matrix-vector formula, which can be seen as a generic way of proving updatable VLR in zero-knowledge.

Finally, by combining the modified ZK system in Sect. 4.2 with a KE protocol [18], we obtain the desired algorithm $\mathsf{Handshake}$, where participants can anonymously authenticate each other and negotiate a session key.

5.1 Description of the Scheme

As in [12,26,28], we imagine a binary tree of depth $d = \log T$ where the root has tag ϵ. For a node at depth $\leq d$ with tag w, its left and right children have tags $w0$ and $w1$, respectively. Lifetime of our scheme is divided into $T = 2^d$ discrete periods, such that successive periods $t \in [T]$ are associated with leaves of the binary tree in the LTR order. To derive keys from previous periods, let $\mathsf{Evolve}_{(t \to T-1)}$ be the set containing exactly one ancestor of each leaf or the leaf itself between period t and $T-1$. This set can be determined by function $\mathsf{sibling}$ in [12] or algorithm $\mathsf{NodeSelect}$ in [26]. Our FSSH scheme is described as follows.

- Setup. Given a security parameter $\lambda \in \mathbb{N}$, this algorithm specifies the following:
  - Maximum member size of a group $N = 2^\ell$, time period bound $T = 2^d$.

- Integer $n = \mathcal{O}(\lambda)$, prime modulus $q = \widetilde{\mathcal{O}}(n^2) > T$, $k = \lceil \log q \rceil$ and dimension $m = 2nk, \overline{m} = nk$. B-bounded distribution χ over $\mathbb{Z}$ with $B = \sqrt{n}\omega(\log n)$.
- Discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ with parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$. Let $\beta = \lceil \sigma \cdot \log n \rceil$ be the upper bound of samples from $D_{\mathbb{Z},\sigma}$.
- Guassian parameters $\overline{\sigma}_i = m^{\frac{3}{2}i+\frac{1}{2}} \cdot \omega(\log^{2k} n)$ for $i \in [d]$, and $\sigma_d = \overline{\sigma}_d \sqrt{m}\omega(\sqrt{\log m})$. Integer bound $\beta_d = \lceil \sigma_d \cdot \log n \rceil$.
- Uniformly random vector $\mathbf{u}_0 \in \mathbb{Z}_q^n$ and matrices $\mathbf{R}_i^b \leftarrow \mathsf{SampleR}(1^m)$ for all $i \in [d]$ and $b \in \{0,1\}$, $\mathbf{Q} = [\mathbf{Q}_1|\mathbf{Q}_2] \in \mathbb{Z}_q^{n\times(nk+d)}$, $\mathbf{F} \in \mathbb{Z}_q^{2n\times nmk}$, $\mathbf{K} \in \mathbb{Z}_{q_1}^{n_1\times n_1}$.
- Matrix dimensions $n_1 = \mathsf{poly}(\lambda), m_1 = \mathcal{O}(n_1)$, integer modulus $q_1 = 2^{\mathcal{O}(n_3)}$, and integer $\theta \geq \frac{2\lambda}{n_1 m_1}$ for session key exchange.
- Discrete Gaussian distribution χ_1 over $\mathbb{Z}$ with deviation $\sigma_1 > \sqrt{\frac{2n_1}{\pi}}$.
- Injective mapping $F : \mathbb{Z}_{q_1}^{n_1\times m_1} \rightarrow [-p,p]^t$ and its inverse F^{-1}, where p, t are defined in [41]. Random oracle $\mathcal{H}_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q^{m\times n}$ and collision resistant hash function $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.

Outputs global public parameter

$$\mathsf{par} = \{N, \ell, T, d, n, q, k, m, \overline{m}, \chi, B, \sigma, \beta, \{\overline{\sigma}_k\}_{k=1}^d, \sigma_d, \beta_d, \mathbf{R}_1^0, \mathbf{R}_1^1, \ldots, \mathbf{R}_d^0, \mathbf{R}_d^1,$$
$$\mathbf{Q}, \mathbf{F}, n_1, m_1, q_1, \theta, \chi_1, \sigma_1, \mathbf{u}_0, \mathbf{K}, F, F^{-1}, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2\}.$$

- $\mathsf{CreateGroup}$. On input par, GA performs the following to establish a new group.
  1. Run $\mathsf{TrapGen}(n,m,q)$ to get a tuple $(\mathbf{G}, \mathbf{T_G})$, then sample matrices $\mathbf{G}_0, \mathbf{G}_1, \mathbf{D}_0, \mathbf{D}_1 \leftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D} \leftarrow U(\mathbb{Z}_q^{n\times\overline{m}})$ and vector $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$.
  2. Run $\mathsf{TrapGen}(n,m,q)$ to generate a tracing key pair $(\mathbf{B}, \mathbf{S})$ (assume all groups share the same tracing keys).
  3. Set registration table $\mathsf{reg} = \emptyset$ and secret key $\mathsf{gsk} = (\mathbf{T_G}, \mathbf{S})$; Publish group public key $\mathsf{gpk} = (\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{B}, \mathbf{u})$ and revocation list $\mathsf{RL} = \emptyset$.
- $\mathsf{AddMember}$. At time period t, one prospective user U_i and GA interact in the following protocol to enroll her in group G. Denote $\mathbf{t} = (\mathbf{t}[1], \ldots, \mathbf{t}[d])$ as the binary representation of t with length d hereunder.
  1. U_i runs $\mathsf{TrapGen}(n,m,q)$ to generate a pair $(\mathbf{A}_i, \mathbf{T}_i)$, and builds the set $\mathsf{Evolve}_{(t\rightarrow T-1)}$. For $\mathbf{s} \in \mathsf{Evolve}_{(t\rightarrow T-1)}$, if $\mathbf{s} = \perp$, set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \perp$. Otherwise, denote $d_{\mathbf{s}}$ as the length of $\mathbf{s}$ holding $d_{\mathbf{s}} \leq d$, set matrix $\mathbf{R}^{(\mathbf{s})} = (\mathbf{R}_1^{\mathbf{s}[1]})^{-1} (\mathbf{R}_2^{\mathbf{s}[2]})^{-1} \ldots (\mathbf{R}_{d_s}^{\mathbf{s}[d_s]})^{-1} \in \mathbb{Z}_q^{n\times m}$, and proceed as follows:
     a) If $d_{\mathbf{s}} = d$, compute a short vector $\mathbf{v}_{i\|\mathbf{s}}$ via $\mathsf{SamplePre}(\mathbf{A}_i\mathbf{R}^{(\mathbf{s})}, \mathsf{BasisDel}(\mathbf{A}_i, (\mathbf{R}^{(\mathbf{s})})^{-1}, \mathbf{T}_i, \overline{\sigma}_d), \mathbf{u}, \sigma_d)$. Set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \mathbf{v}_{i\|\mathbf{s}}$.
     b) Else, evaluate $\mathsf{BasisDel}(\mathbf{A}_i, (\mathbf{R}^{(\mathbf{s})})^{-1}, \mathbf{T}_i, \overline{\sigma}_{d_{\mathbf{s}}})$ to obtain a short basis $\mathbf{T}_{i\|\mathbf{s}}$ for $\Lambda_q^{\perp}(\mathbf{A}_i\mathbf{R}^{(\mathbf{s})})$, and set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \mathbf{T}_{i\|\mathbf{s}}$.

     Now let $\mathsf{upk}_i = \mathbf{A}_i$ be the long-term public key and $\mathsf{usk}_{i\|t} = \{\mathsf{usk}_{i\|t}[\mathbf{s}] \mid \mathbf{s} \in \mathsf{Evolve}_{(t\rightarrow T-1)}\}$ be the initial secret key of U_i. Finally, U_i samples a proof vector $\mathbf{v}_i \leftarrow \mathsf{SamplePre}(\mathbf{A}_i, \mathbf{T}_i, \mathbf{u}_0, \sigma_d)$ and sends $(\mathbf{A}_i, \mathbf{v}_i)$ to GA. She discards the original $\mathbf{T}_i$ for forward security.

2. Upon receiving the request from U_i, GA first checks: (i) whether there is a collision between $\mathbf{A}_i$ and the previous records of users' public keys; (ii) whether $\mathbf{A}_i$ is valid w.r.t. $\mathbf{v}_i$ by verifying $\mathbf{A}_i\mathbf{v}_i = \mathbf{u}_0$ and $\|\mathbf{v}_i\|_\infty \leq \beta_d$. If either case occurs, GA outputs $\perp$ and aborts. Otherwise, GA performs the following steps to issue a credential to U_i.
   a) To generate user's revocation token, set $\mathsf{urt}_{i\|t} = \mathbf{Q}_1 \cdot \mathsf{vdec}_{n,q-1}(\mathbf{a}_{i,0}) + \mathbf{Q}_2 \cdot \mathbf{t} \in \mathbb{Z}_q^n$, where $\mathbf{a}_{i,0}$ is the first column of $\mathbf{A}_i$ and we assume that it is a non-zero vector. In the long term, the current time period is embedded in the corresponding token, such that no adversary can deploy a previous token to conduct a handshake.
   b) Choose a random spare $\mathbf{i} \in \{0,1\}^\ell$ for user's identity ID_i having decimal value i. Then hash U_i's public key as $\mathbf{h}_i = \mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{A}_i^\top) \in \mathbb{Z}_q^{2n}$.
   c) Encode the identity through building the compressed matrix $\mathbf{G}^{(i)} = [\mathbf{G}|\mathbf{G}_0 + i \cdot \mathbf{G}_1]$. Runs $\mathsf{ExtBasis}(\mathbf{G}^{(i)}, \mathbf{T}_\mathbf{G})$ to get a basis $\mathbf{T}_\mathbf{G}^{(i)}$ for $\mathbf{G}^{(i)}$.
   d) Sample $\mathbf{r}_i \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, and compute the chameleon hash of $\mathbf{h}_i$ as $\mathbf{c}_i = \mathbf{D}_0 \cdot \mathbf{r}_i + \mathbf{D}_1 \cdot \mathsf{vdec}_{2n,q-1}(\mathbf{h}_i)$.
   e) Invoke $\mathsf{SamplePre}(\mathbf{G}^{(i)}, \mathbf{T}_\mathbf{G}^{(i)}, \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_i), \sigma)$ to obtain a short vector $\mathbf{d}_i \in \mathbb{Z}^{2m}$ satisfying that

$$\mathbf{G}^{(i)}\mathbf{d}_i = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_i) \mod q, \tag{4}$$

   then return the credential $\mathsf{cred}_i = (\mathsf{upk}_i, \mathsf{ID}_i, \mathsf{urt}_{i\|t}, \mathbf{d}_i, \mathbf{r}_i)$ to U_i and adds cred_i to table reg.
3. U_i verifies that cred_i is consistent with Eq. 4 and $\mathbf{d}_i \in [-\beta, \beta]^{2m}$, $\mathbf{r}_i \in [-\beta, \beta]^m$. She aborts if it is not the case. To avoid confusion, use $t_{i,add} = t$ to denote the time period at which U_i has been registered.

- $\mathsf{Update}_\mathsf{U}$. At the beginning of time period t, member U_i conducts the following procedures to update her secret pair $(\mathsf{cred}_{i\|t-1}, \mathsf{usk}_{i\|t-1})$.

  For revocation token update, compute $\mathsf{urt}_{i\|t} = \mathbf{Q}_1 \cdot \mathsf{vdec}_{n,q-1}(\mathsf{urt}_{i\|t-1}) + \mathbf{Q}_2 \cdot \mathbf{t} \in \mathbb{Z}_q^n$. W.L.O.G, we assume that there is no all-zero token or two identical tokens. (Otherwise, the user would find a solution to $\mathsf{SIS}_{n,q,\sqrt{nk+d}}$ problem associated with matrix $\mathbf{Q}$, which is of negligible probability.)

  For the secret key derivation, first specify the node set $\mathsf{Evolve}_{(t\to T-1)}$. Then for $\mathbf{s} \in \mathsf{Evolve}_{(t\to T-1)}$, if $\mathbf{s} = \perp$, set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \perp$, otherwise, there exists exactly one $\mathbf{s}' \in \mathsf{Evolve}_{(t-1\to T-1)}$ as the prefix of $\mathbf{s}$, i.e., $\mathbf{s} = \mathbf{s}'\|\mathbf{x}$ for some binary string $\mathbf{x}$. Consider two cases:
  1. If $\mathbf{s} = \mathbf{s}'$, set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \mathsf{usk}_{i\|t-1}[\mathbf{s}']$.
  2. Else, it holds that $\mathbf{x}$ is not empty and $\mathsf{usk}_{i\|t-1}[\mathbf{s}'] = \mathbf{T}_{i\|\mathbf{s}'}$ is a short basis. Compute matrix $\mathbf{R}^{(\mathbf{x})} = (\mathbf{R}_{1+d_{\mathbf{s}'}}^{\mathbf{x}[1]})^{-1}\, (\mathbf{R}_{2+d_{\mathbf{s}'}}^{\mathbf{x}[2]})^{-1} \ldots (\mathbf{R}_{d_\mathbf{s}}^{\mathbf{x}[d_\mathbf{x}]})^{-1}$, then consider the following two sub-cases:
     a) If $d_\mathbf{s} = d$, generate a short vector $\mathbf{v}_{i\|\mathbf{s}}$ by running $\mathsf{SamplePre}(\mathbf{A}_i\mathbf{R}^{(\mathbf{s}')}\mathbf{R}^{(\mathbf{x})}, \mathsf{BasisDel}(\mathbf{A}_i\mathbf{R}^{(\mathbf{s}')}, (\mathbf{R}^{(\mathbf{x})})^{-1}, \mathbf{T}_{i\|\mathbf{s}'}, \overline{\sigma}_{d_\mathbf{s}}), \mathbf{u}, \sigma_d)$. Set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \mathbf{v}_{i\|\mathbf{s}}$.

b) If $d_{\mathbf{s}} < d$, run $\mathsf{BasisDel}(\mathbf{A}_i\mathbf{R}^{(\mathbf{s}')}, (\mathbf{R}^{(\mathbf{x})})^{-1}, \mathbf{T}_{i\|\mathbf{s}'}, \overline{\sigma}_{d_{\mathbf{s}}})$ to obtain a short basis $\mathbf{T}_{i\|\mathbf{s}}$, and set $\mathsf{usk}_{i\|t}[\mathbf{s}] = \mathbf{T}_{i\|\mathbf{s}}$.

Set $\mathsf{usk}_{i\|t} = \{\mathsf{usk}_{i\|t}[\mathbf{s}] \mid \mathbf{s} \in \mathsf{Evolve}_{(t\to T-1)}\}$ and erases the previous one.

- Handshake. At time period t, suppose a member A from group G_a with $\mathsf{gpk}_a = (\mathbf{G}^{(a)}, \mathbf{G}_0^{(a)}, \mathbf{G}_1^{(a)}, \mathbf{D}^{(a)}, \mathbf{D}_0^{(a)}, \mathbf{D}_1^{(a)}, \mathbf{B}, \mathbf{u}_a)$, $\mathsf{cred}_a = (\mathsf{upk}_a, \mathsf{ID}_a, \mathsf{urt}_{a\|t}, \mathbf{d}_a, \mathbf{r}_a)$, revocation list RL_a, $\mathsf{usk}_{a\|t} = \{\mathsf{usk}_{a\|t}[\mathbf{s}] \mid \mathbf{s} \in \mathsf{Evolve}_{t\to T-1}\}$, and another member B from group G_b having $(\mathsf{gpk}_b, \mathsf{cred}_b, \mathsf{RL}_b, \mathsf{usk}_{b\|t})$ of same structure, aim to execute a handshake. They proceed the following two-round protocol.
  1. $A \to B : (\mathtt{PROOF}_a)$
     a) A samples a small private key $\mathbf{S}_a \hookleftarrow \chi_1(\mathbb{Z}_{q_1}^{n_1\times m_1})$ and a small noise $\mathbf{E}_a \hookleftarrow \chi_1(\mathbb{Z}_{q_1}^{n_1\times m_1})$. Then she computes $\mathbf{C}_a = \mathbf{K}\cdot\mathbf{S}_a + \mathbf{E}_a \in \mathbb{Z}_{q_1}^{n_1\times m_1}$.
     b) Parse $\mathsf{upk}_a = \mathbf{A}_a$, A fetches the secret key for string $\mathbf{t}$ from $\mathsf{usk}_{a\|t}$ as $\mathbf{v}_{a\|\mathbf{t}}$ and assembles the corresponding matrix $\mathbf{A}_{a\|t}$ as
$$\mathbf{A}_{a\|t} = \mathbf{A}_a\,(\mathbf{R}_1^{\mathbf{t}[1]})^{-1}\,(\mathbf{R}_2^{\mathbf{t}[2]})^{-1}\ldots(\mathbf{R}_d^{\mathbf{t}[d]})^{-1} \in \mathbb{Z}_q^{n\times m}. \tag{5}$$
     c) A samples $\rho_a \xleftarrow{\$} \{0,1\}^n$ and let $\mathbf{W}_a = \mathcal{H}_0(\mathsf{gpk}_a, \rho_a)$. Next, she computes $\mathbf{w}_a = \mathbf{W}_a \cdot \mathsf{urt}_{a\|t} + \mathbf{e}_a \mod q$ where $\mathbf{e}_a \hookleftarrow \chi^m$.
     d) A samples $\mathbf{P}_a \leftarrow U(\mathbb{Z}_q^{n\times\ell})$, $\mathbf{s}^{(a)} \hookleftarrow \chi^n$, $\mathbf{e}_1^{(a)} \hookleftarrow \chi^m$, $\mathbf{e}_2^{(a)} \hookleftarrow \chi^\ell$, so that produces the ciphertext $(\mathbf{c}_1^{(a)}, \mathbf{c}_2^{(a)})$ as
$$(\mathbf{c}_1^{(a)} = \mathbf{B}^\top\cdot\mathbf{s}^{(a)} + \mathbf{e}_1^{(a)}, \mathbf{c}_2^{(a)} = \mathbf{P}_a^\top\cdot\mathbf{s}^{(a)} + \mathbf{e}_2^{(a)} + \lfloor\frac{q}{2}\rfloor\cdot\mathsf{ID}_a). \tag{6}$$
     e) With public input $\mathsf{pp}_a = (\mathsf{par}, \mathsf{gpk}_a, \mathbf{c}_1^{(a)}, \mathbf{c}_2^{(a)}, t_a)$ where $t_a = t - t_{a,add}$, A runs $\mathtt{Prove}_{\mathtt{hs}}$ designed in Sect. 4.2 to generate a proof π_a for $\xi_a = (\mathsf{ID}_a, \mathsf{urt}_{a\|t}, \mathbf{d}_a, \mathbf{r}_a, \mathbf{A}_a, \mathbf{v}_{a\|\mathbf{t}}, \mathbf{e}_a, \mathbf{s}^{(a)}, \mathbf{e}_1^{(a)}, \mathbf{e}_2^{(a)})$, satisfying that:
        - $\mathsf{urt}_{a\|t}$ is correctly derived from $\mathbf{A}_a$ after t_a times of updates.
        - $(\mathsf{ID}_a, \mathbf{d}_a, \mathbf{r}_a)$ satisfies Eq. 4 with the specific form in AddMember.
        - $\mathbf{W}_a \cdot \mathsf{rt}_a + \mathbf{e}_a = \mathbf{w}_a$ and $\|\mathbf{e}_a\|_\infty \le B$.
        - Eq. 5 holds with $\mathbf{A}_{a\|t}\cdot\mathbf{v}_{a\|\mathbf{t}} = \mathbf{u}_a \mod q$ and $\|\mathbf{v}_{a\|\mathbf{t}}\|_\infty \le \beta_d$.
        - Eq. 6 holds with $\|\mathbf{s}^{(a)}\|_\infty \le B$, $\|\mathbf{e}_1^{(a)}\|_\infty \le B$, and $\|\mathbf{e}_2^{(a)}\|_\infty \le B$.

     Note that the challenge part of π_a is modified as $\widetilde{ch_a} := ch_a \oplus F(\boldsymbol{C}_a)$.
     f) A finally sends $\mathtt{PROOF}_a = (\rho_a, \mathbf{w}_a, \mathbf{P}_a, \mathbf{c}_1^{(a)}, \mathbf{c}_2^{(a)}, t_a, \pi_a)$ to B.
  2. $B \to A : (\mathtt{PROOF}_b, \mathtt{V}_b)$
     a) B computes $\mathbf{W}'_a = \mathcal{H}_0(\mathsf{gpk}_b, \rho_a)$. Then he checks if there exists an index i such that $\mathbf{e}'_i = \mathbf{w}_a - \mathbf{W}'_a\cdot\mathbf{v}_i$ and $\|\mathbf{e}'_i\|_\infty \le B$ for $\mathbf{v}_i \in \mathsf{RL}_b$. If so, B sends A a random pair $(\mathtt{PROOF}_b, \mathtt{V}_b)$ and aborts. Otherwise, he continues to perform the following steps.
     b) B runs $\mathtt{Verify}_{\mathtt{hs}}(\mathsf{pp}'_a, \pi_a)$ where $\mathsf{pp}'_a = (\mathsf{par}, \mathsf{gpk}_b, \mathbf{c}_1^{(a)}, \mathbf{c}_2^{(a)}, t_a)$, to recover the hidden message of A as $\mathbf{C}'_a = F^{-1}(ch'_a \oplus \widetilde{ch_a})$.
     c) B samples his ephemeral private key $\mathbf{S}_b \hookleftarrow \chi_1(\mathbb{Z}_{q_1}^{n_1\times m_1})$ and a small noise $\mathbf{E}_b \hookleftarrow \chi_1(\mathbb{Z}_{q_1}^{n_1\times m_1})$. Then B computes $\mathbf{C}_b = \mathbf{K}\cdot\mathbf{S}_b + \mathbf{E}_b$.

   d) Similarly, B computes the LWE function of his revocation token as $\mathbf{w}_b = \mathbf{W}_b \cdot \mathsf{urt}_{b\|t} + \mathbf{e}_b$, and encrypts his identity as $(\mathbf{c}_1^{(b)}, \mathbf{c}_2^{(b)})$.
   e) With analogous public input pp_b, B runs $\mathtt{Prove}_{\mathtt{hs}}$ to generate an argument π_b for his secret tuple ξ_b, of which each element meets the similar constraints as that of ξ_a. Remark $\widetilde{ch_b} := ch_b \oplus F(\boldsymbol{C}_b^\top)$.
   f) Upon obtaining $\mathbf{C}'_a$, B generates the check matrix $\mathbf{M}$ and the communication key $\mathbf{K}_b$ as depicted in Sect. 2.4. Then B computes the authentication code $\mathtt{V}_b = \mathcal{H}_1(\mathbf{K}_b \| \mathbf{C}_b \| \mathbf{0})$.
   g) B dispatches $\mathtt{PROOF}_b = (\rho_b, \mathbf{w}_b, \mathbf{P}_b, \mathbf{c}_1^{(b)}, \mathbf{c}_2^{(b)}, t_b, \pi_b, \mathbf{M})$ and $\mathtt{V}_b$ to A.
3. $A \rightarrow B : (\mathtt{V}_a)$
   a) A computes $\mathbf{W}'_b = \mathcal{H}_0(\mathsf{gpk}_a, \rho_b)$ and also checks if there exists an index j such that $\mathbf{e}'_j = \mathbf{w}_b - \mathbf{W}'_b \cdot \mathbf{v}_j$ and $\|\mathbf{e}'_j\|_\infty \leq B$ for $\mathbf{v}_j \in \mathsf{RL}_a$. If so, A responds a random value $\mathtt{V}_a \leftarrow U(\{0,1\}^{q_1}$, outputs 0 and aborts. Otherwise, A moves to execute the following steps.
   b) A also runs $\mathtt{Verify}_{\mathtt{hs}}(\mathsf{pp}'_b, \pi_b)$ to retrieve $\mathbf{C}'^\top_b = F^{-1}(\widetilde{ch_b} \oplus ch'_b)$.
   c) A extracts the shared key $\mathbf{K}_a$ following the steps in Sect. 2.4. Then A verifies that $\mathtt{V}_b \overset{?}{=} \mathcal{H}_1(\mathbf{K}_a \| \mathbf{C}'_b \| \mathbf{0})$. If so, A outputs 1 and sends $\mathtt{V}_a = \mathcal{H}(\mathbf{K}_a \| \mathbf{C}_a \| \mathbf{1})$ to B. Else, A outputs 0 and responds a random $\mathtt{V}_a$.
   d) B verifies $\mathtt{V}_a$ via a similar equation $\mathtt{V}_a \overset{?}{=} \mathcal{H}_1(\mathbf{K}_b \| \mathbf{C}'_a \| \mathbf{1})$. B outputs 1 if the equation holds, else he outputs 0.

- $\mathsf{TraceMember}$. With transcript $(\mathtt{PROOF}, \mathtt{V})$ of a handshake executed at time period t, TA performs the following steps to trace the involved group member:
  1. Parse $\mathtt{PROOF} = (\rho, \mathbf{w}, \mathbf{P}, \mathbf{c}_1, \mathbf{c}_2, t^*, \pi)$ where $\mathbf{P} = [\mathbf{p}_1|\ldots|\mathbf{p}_\ell] \in \mathbb{Z}_q^{n\times\ell}$. Then for all $i \in [\ell]$, invoke $\mathsf{SamplePre}(\mathbf{B}, \mathbf{S}, \mathbf{p}_i, \sigma)$ to obtain a small vector $\mathbf{f}_i$. Set $\mathbf{F} = [\mathbf{f}_1|\ldots|\mathbf{f}_\ell]$ such that $\mathbf{B} \cdot \mathbf{F} = \mathbf{P} \mod q$.
  2. Decrypt $(\mathbf{c}_1, \mathbf{c}_2)$ by computing $\mathsf{ID} = \lfloor \mathbf{c}_2 - \mathbf{F}^\top \cdot \mathbf{c}_1 / \lfloor q/2 \rfloor \rfloor \in \{0,1\}^\ell$.
  3. If there exists an elements $\mathsf{ID}_i = \mathsf{ID}$, return ID_i. Otherwise, output $\perp$.
- $\mathsf{RemoveMember}$. To remove ID_i from group G at the beginning of period t, GA gets the initial revocation token $\mathsf{urt}_{i\|t_{i,add}}$ of ID_i from table Reg, and adds it to public list RL. Since the current token can be computed from the initial one, for simplicity we assume the elements of RL are all the updated ones.

5.2 Analysis of the Scheme

Completeness. We demonstrate that our scheme is complete with overwhelming probability if A and B belong to the same group $(\mathsf{gpk}_1 = \mathsf{gpk}_2)$ with unrevoked and updated group secret, and they both follow the specified protocol.

First, by *completeness* of system Π_{hs}, both A and B can produce a valid proof (π_a, π_b) at the first round of a handshake, which means that the receiver can always rightly recover the original challenge $ch' = ch$, such that they can retrieve the hidden message $\mathbf{C}' = \mathbf{C}$. It follows that $\mathbf{K}_a \neq \mathbf{K}_b$ with negligible probability from Theorem 2. Therefore, the two members can verify the corresponding equations successfully, i.e., the message authentication code $\mathtt{V}_a$ $(\mathtt{V}_b)$

is correct. Consequently, the handshake protocol will output 1 for both participants.

Next, we show that $\mathsf{TraceMember}$ always outputs ID_a (ID_b). Observe that, the decryption procedure computes $\mathbf{c}_2 - \mathbf{F}^\top \cdot \mathbf{c}_1 = \mathbf{P}^\top \cdot \mathbf{s} + \mathbf{e}_2 + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{ID} - \mathbf{F}^\top \cdot (\mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}_1)$, which can be further simplified into $\mathbf{e}_2 - \mathbf{F}^\top \cdot \mathbf{e}_1 + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{ID}$, where $\|\mathbf{e}_1\|_\infty \leq B$, $\|\mathbf{e}_2\|_\infty \leq B$, and $\|\mathbf{f}_i\|_\infty \leq \lceil \sigma \cdot \log m \rceil$ implied by Lemma 1. Recall that $q = \widetilde{\mathcal{O}}(n^2)$, $m = 2n \log q$ and $B = \sqrt{n}\omega(\log n)$. Hence we always have $\|\mathbf{e}_2 - \mathbf{F}^\top \cdot \mathbf{e}_1\|_\infty \leq B + m \cdot B \cdot \lceil \sigma \cdot \log m \rceil < q/5$, inducing that the $\mathsf{ID} = \mathsf{ID}_a$.

Security. Now we give security analysis for our scheme, proof of the following theorem is deferred to Appendix A.

Theorem 3. *In the random oracle model, our scheme satisfies the forward impersonator resistance, detector resistance and backward unlinkability under the* SIS, ISIS *and* LWE *assumptions.*

Efficiency. Finally we analyze the complexity of our scheme, with respect to security parameter λ, two system parameters $\ell = \log N$ and $d = \log T$.

- Group public key contains several matrices and a vector with bit-size $\widetilde{\mathcal{O}}(\lambda^2)$.
- Group credential consists of 4 vectors and has bit-size $\widetilde{\mathcal{O}}(\lambda + \ell)$.
- User secret key has one vector from $\mathsf{SamplePre}$ and at most d trapdoor matrices from $\mathsf{BasisDel}$, all of which have bit-size $\widetilde{\mathcal{O}}(\lambda^2 d^3)$.
- The communication cost of a handshake protocol can be viewed as four parts: $(\rho, \mathbf{w}, \mathbf{P}, t)$ for revocation check; Modified ZK argument π_{hs}, whose bit-size largely relies on the length of witness $\mathbf{x}$ and can be quantized as $\widetilde{\mathcal{O}}(\lambda^2 + d \cdot \lambda + \ell^2)$; Two IBE ciphertexts and one authentication code. Overall, the dispatched data has bit-size $\widetilde{\mathcal{O}}(\lambda^2 + (d + \ell) \cdot \lambda + \ell^2)$.
- Dynamic revocation list has bit-size $\widetilde{\mathcal{O}}(N \cdot \lambda)$.

Table 1. Comparison between scheme [5] and ours.

Scheme	gpk	cred	usk	Handshake cost	RL	FS
[5]	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\ell \cdot \lambda)$	$\widetilde{\mathcal{O}}(\ell \cdot \lambda)$	$\widetilde{\mathcal{O}}(\ell \cdot \lambda)$	$\widetilde{\mathcal{O}}(N \cdot \lambda)$	✗
Ours	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda + \ell)$	$\widetilde{\mathcal{O}}(\lambda^2 d^3)$	$\widetilde{\mathcal{O}}(\lambda^2 + (d + \ell) \cdot \lambda + \ell^2)$	$\widetilde{\mathcal{O}}(N \cdot \lambda)$	✓

In Table 1, we give a detailed comparison of our scheme with the only known lattice-based one [5], in terms of efficiency and functionality. Note that forward security is achieved with a reasonable increase in communication cost, thanks to the more efficient ZK system [41]. Besides, our scheme allows dynamic user enrollment. In other words, users autonomously generate their secret keys rather than being issued by GA, which prevents malicious GA from framing honest users.

Acknowledgements. This work is supported by Guangdong Major Project of Basic and Applied Basic Research (2019B030302008) and the National Natural Science Foundation of China (No. 61972429) and Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515011797) and the Opening Project of Guangdong Provincial Key Laboratory of Information Security Technology (2020B1212060078-09).

A Deferred Proof of Theorem 3

Proof. We prove Theorem 3 by separately proving that our scheme satisfies the 3 required properties defined in Sect. 3.

Forward Impersonator Resistance. We prove this property by contradiction. Suppose that a PPT adversary $\mathcal{A}$ succeeds in experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{F\text{-}IR}}$ with non-negligible advantage ϵ. Then we can build a PPT algorithm $\mathcal{B}$ that solves $\mathsf{SIS}_{n,m,q,2\sqrt{m}\beta_d}$ problem with non-negligible probability.

Given an SIS instance $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the goal of $\mathcal{B}$ is to find a non-zero vector $\mathbf{z} \in \mathbb{Z}_q^m$ such that $\mathbf{A}\cdot\mathbf{z} = \mathbf{0} \mod q$ and $\|\mathbf{z}\| \le \sqrt{m}\beta$. Towards this goal, $\mathcal{B}$ first prepares a simulated attack environment for $\mathcal{A}$ as follows:

- Randomly guess the target user's identity $\mathsf{ID}^* : \mathbf{i}^* \in \{0,1\}^\ell$ and forgery time period $t^* \in [0, T-1]$.
- Sample random matrices $\mathbf{R}_1^{\mathbf{t}^*[1]}, \mathbf{R}_2^{\mathbf{t}^*[2]}, \ldots, \mathbf{R}_d^{\mathbf{t}^*[d]} \in \mathbb{Z}^{m\times m}$ from the distribution $\mathcal{D}_{m\times m}$. Set $\mathbf{A}_{i^*} = \mathbf{A}\,\mathbf{R}_d^{\mathbf{t}^*[d]}\cdots\mathbf{R}_2^{\mathbf{t}^*[2]}\,\mathbf{R}_1^{\mathbf{t}^*[1]} \in \mathbb{Z}_q^{n\times m}$, which is the public key of target user ID^*.
- Sample $\mathbf{v} \hookleftarrow D_{\mathbb{Z}^m,\sigma_d}$. If $\|\mathbf{v}\|_\infty > \beta_d$, then repeat the sampling. Compute $\mathbf{u}^* = \mathbf{A}\cdot\mathbf{v} \mod q$.
- Assemble d matrices $\mathbf{F}_j = \mathbf{A}_{i^*}\,(\mathbf{R}_1^{\mathbf{t}^*[1]})^{-1}\ldots(\mathbf{R}_j^{\mathbf{t}^*[j]})^{-1}$ for $j \in [0, d-1]$ ($\mathbf{F}_0 = \mathbf{A}_{i^*}$). For each $\mathbf{F}_j$, invoke $\mathsf{SampleRwithBasis}(\mathbf{F}_j)$ to obtain a matrix $\mathbf{R}_{j+1}^{1-\mathbf{t}^*[j+1]}$, along with a short basis $\mathbf{T}_{j+1}$ for $\Lambda^\perp(\mathbf{F}'_{j+1})$ where $\mathbf{F}'_{j+1} = \mathbf{F}_j\,(\mathbf{R}_{j+1}^{1-\mathbf{t}^*[j+1]})^{-1}$. As the simulation in [2], $\mathcal{B}$ can use these bases to generate ID^*'s secret key for every period $t' > t^*$.
- Generate other elements of $(\mathsf{gpk}^*, \mathsf{gsk}^*)$ for group G^* that ID^* belongs to.
- Operates as GA in algorithm $\mathsf{AddMember}$ to determine the target user's credential $\mathsf{cred}_{\mathsf{ID}^*\|t^*}$ at period t^*.

Note that, by construction, the distribution of $(\mathsf{par}^*, \mathsf{gpk}^*, \mathsf{gsk}^*, \mathsf{cred}_{\mathsf{ID}^*\|t^*})$ is statistically close to that of the real scheme, and the choice of (ID^*, t^*) is hidden from the adversary.

$\mathcal{B}$ responds to $\mathcal{A}$'s queries of $\{\mathsf{KeyP}, \mathsf{Trace}, \mathsf{Remove}, \mathsf{AddU}, \mathsf{KeyG}\}$ exactly the same as the real scheme. For other queries at current period t, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

- When $\mathcal{A}$ queries random oracles $\mathcal{H}_0$ or $\mathcal{G}$, $\mathcal{B}$ replies with uniformly random strings and records the inputs/outputs of these queries.
- For queries of oracle CorU, if the requested user has been already corrupted, i.e., $(\mathsf{ID}, \cdot, \cdot, \ ) \in \mathsf{CU}$, $\mathcal{B}$ aborts. Otherwise, consider two cases:

i) The chosen user's identity is ID^*. If $t \le t^*$, $\mathcal{B}$ aborts. Otherwise, for each node $\mathbf{s} \in \mathsf{Evolve}_{t\to T-1}$, denote the length of $\mathbf{s}$ as $d_{\mathbf{s}}$, $\mathcal{B}$ first computes the smallest index $j_{\mathbf{s}}$ such that $1 \le j_{\mathbf{s}} \le d_{\mathbf{s}}$ and $\mathbf{s}[j_{\mathbf{s}}] \ne \mathbf{t}^*[j_{\mathbf{s}}]$. After setting delegation matrix $\mathbf{R}^{(\mathbf{s})} = (\mathbf{R}_{j_{\mathbf{s}}+1}^{\mathbf{s}[j_{\mathbf{s}}+1]})^{-1} \cdots (\mathbf{R}_{d_{\mathbf{s}}}^{\mathbf{s}[d_{\mathbf{s}}]})^{-1}$, $\mathcal{B}$ computes $\mathsf{usk}_{i^*\|t}[\mathbf{s}]$ via $\mathsf{SamplePre}(\mathbf{F}'_{j_{\mathbf{s}}}\mathbf{R}^{(\mathbf{s})}, \mathsf{BasisDel}(\mathbf{F}'_{j_{\mathbf{s}}}, (\mathbf{R}^{(\mathbf{s})})^{-1}, \mathbf{T}_{j_{\mathbf{s}}}, \overline{\sigma}_{d_{\mathbf{s}}}), \mathbf{u}, \sigma_d)$ if $d_{\mathbf{s}} = d$, or via $\mathsf{BasisDel}(\mathbf{F}'_{j_{\mathbf{s}}}, (\mathbf{R}^{(\mathbf{s})})^{-1}, \mathbf{T}_{j_{\mathbf{s}}}, \overline{\sigma}_{d_{\mathbf{s}}})$ if $d_{\mathbf{s}} < d$. Next, $\mathcal{B}$ builds $\mathsf{usk}_{i^*\|t}$ and derives $\mathsf{cred}_{i^*\|t}$ as in our main scheme. Finally $\mathcal{B}$ returns the secret pair to $\mathcal{A}$ and adds (ID^*, G^*, t) to CU. Note that $\mathcal{A}$ can not obtain the target user's secret until period $t^* + 1$.

ii) $\mathsf{ID} \ne \mathsf{ID}^*$, then $\mathcal{B}$ can perfectly answer the query as it stores the initial secret key (a short basis $\mathbf{T}$) when ID was enrolled in group G. In other words, $\mathcal{B}$ performs as that in $\mathsf{Update}_{\mathsf{U}}$ to derive user's secret pair $(\mathsf{cred}_{\mathsf{ID}\|t}, \mathsf{usk}_{\mathsf{ID}\|t})$ and returns it to $\mathcal{A}$. Finally, it adds (ID, G, t) to CU.

- For queries of oracle HS with input ID, if $(\mathsf{ID}, \cdot, \cdot) \in \mathsf{CU}$, $\mathcal{B}$ aborts. Otherwise, if $\mathsf{ID} \ne \mathsf{ID}^*$ or $t > t^*$, $\mathcal{B}$ acts as in algorithm $\mathsf{Handshake}$ using the corresponding secrets. Else, $\mathcal{B}$ has to answer without using the user's secret key. To do so, $\mathcal{B}$ also performs the same as in $\mathsf{Handshake}$, except that in the second flow $\mathcal{B}$ generates a simulated proof π' by utilizing the well-designed simulator of applied $\mathsf{NIZKAoK}$ [41].

We claim that $\mathcal{A}$ cannot distinguish whether it interacts with a real challenger or with $\mathcal{B}$. First, the secret pair of ID^* given to $\mathcal{A}$ after period t^* is indistinguishable from the real one, due to the facts that

i) the revocation token is uniform over $\mathbb{Z}_q^n$ and other elements of $\mathsf{cred}_{\mathsf{ID}^*\|t}$ are produced in the same way as that in $\mathsf{AddMember}$;
ii) the outputs of $\mathsf{BasisDel}$ are uniformly random by Lemma 5. Second, the handshake queries make no difference to the view of $\mathcal{A}$, implied by the *zero knowledge* property of the underlying $\mathsf{NIZKAoK}$.

After $\mathcal{A}$ halts with her output $\mathtt{PROOF}^* = (\rho^*, \mathbf{w}^*, \mathbf{P}^*, \mathbf{c}_1^*, \mathbf{c}_2^*, \hat{t}, \pi^*)$ at period t', $\mathcal{B}$ checks if $t' = t^*$. If not, the guess of the impersonator period t^* fails and $\mathcal{B}$ aborts. Else, parse $\pi^* = (cmt_s^*, \widetilde{ch}^*, rsp^*)$, since $\mathcal{A}$ wins, we argue that by *completeness* of our scheme, $\mathcal{A}$ must have queried the related random oracle $\mathcal{G}$ via Fiat-Shamir heuristic on input $\eta^* = (cmt^*, \mathsf{pp}^*)$. Otherwise, guessing correctly this value occurs only with negligible probability $\epsilon' = (\frac{1}{2p+1})^t$. Therefore, with probability at least $\epsilon - \epsilon'$, the tuple η^* has been an input of one hash query, denoted as $\kappa^* \le q_{\mathcal{G}}$, where $q_{\mathcal{G}}$ is the total number of queries to $\mathcal{G}$ made by $\mathcal{A}$.

Next, $\mathcal{B}$ picks κ^* as the target forking point and replays $\mathcal{A}$ polynomial time. For each new run, $\mathcal{B}$ starts with the same random tape and input as in the original execution, but from the κ^*-th query onwards, $\mathcal{B}$ will reply to $\mathcal{A}$ with fresh and independent hash values. Moreover, $\mathcal{B}$ always replies as in the original run for queries of $\mathcal{H}_0$. Note that the input of κ^* hash query must be η^*. The Forking Lemma in [14] implies that, with probability larger than 1/2, $\mathcal{B}$ can obtain 3 forks involving the same tuple η^*, but with pairwise distinct challenges

$$\widetilde{ch}_1^*,\ \widetilde{ch}_2^*,\ \widetilde{ch}_3^* \in [-p, p]^t.$$

Moreover, by the binding property of used commitment scheme, $\mathcal{B}$ can obtain 3 valid tuples from the output of $\mathcal{A}$ as

$$\{(ch_1^*, cmt^*, rsp_1^*), (ch_2^*, cmt^*, rsp_2^*), ch_3^*, cmt^*, rsp_3^*\},$$

by first recovering the unsent cmt_r^* and then the original ch^*. Then by *proof of knowledge* of system Π_{hs}, $\mathcal{B}$ can extract the witness

$$\xi^* = (\mathsf{ID}', \mathsf{urt}^*_{\mathsf{ID}'\|t^*}, \mathbf{d}^*, \mathbf{r}^*, \mathbf{A}_{i^*}, \mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*}, \mathbf{e}^*, \mathbf{s}^*, \mathbf{e}_1^*, \mathbf{e}_2^*),$$

which satisfies that

- $\mathsf{urt}^*_{\mathsf{ID}'\|t^*}$ is correctly derived from $\mathbf{A}_{i^*}$ after $\hat{t}$ times of updates.
- Triple $(\mathsf{ID}', \mathbf{d}^*, \mathbf{r}^*)$ has the specific form as that in algorithm AddMember and satisfies Eq. 4.
- $\mathbf{W}^* \cdot \mathsf{urt}^*_{\mathsf{ID}'\|t^*} + \mathbf{e}^* = \mathbf{w}^*$ and $\|\mathbf{e}^*\|_\infty \le B$, where $\mathbf{W}^* = \mathcal{H}_1(\mathsf{gpk}^*, \rho^*)$.
- $\mathbf{A}_{\mathsf{ID}'\|t^*} = \mathbf{A}_{i^*}\ (\mathbf{R}_1^{\mathbf{t}^*[1]})^{-1}\ (\mathbf{R}_2^{\mathbf{t}^*[2]})^{-1} \dots (\mathbf{R}_d^{\mathbf{t}^*[d]})^{-1}$.
- $\mathbf{A}_{\mathsf{ID}'\|t^*} \cdot \mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*} = \mathbf{u}^* \mod q$ and $\|\mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*}\|_\infty \le \beta_d$.
- $\mathbf{c}_1^* = \mathbf{B}^{*\top} \cdot \mathbf{s}^* + \mathbf{e}_1^*, \mathbf{c}_2^* = \mathbf{P}^{*\top} \cdot \mathbf{s}^* + \mathbf{e}_2^* + \lfloor \frac{q}{2} \rfloor \cdot \mathsf{ID}'$, where $\|\mathbf{s}^*\|_\infty \le B$, $\|\mathbf{e}_1^*\|_\infty \le B$, and $\|\mathbf{e}_2^*\|_\infty \le B$.

Now consider the following cases:

a. There is no element in table reg that contains ID'. This implies that the pair $(\mathbf{A}^*, (\mathsf{ID}', \mathbf{d}^*, \mathbf{r}^*))$ forms a forgery for the SIS-based signature of Sect. 2.2.

b. $\mathsf{ID}' \ne \mathsf{ID}^*$, indicating the guess of the impersonator user fails, then $\mathcal{B}$ aborts.

c. Conditioned on guessing correctly t^* and ID^*, we have that $\mathbf{A}_{\mathsf{ID}'\|t^*} \cdot \mathbf{v}_{\mathsf{ID}'\|t^*} = \mathbf{A} \cdot \mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*} = \mathbf{u}^* \mod q$, recall that $\mathbf{A}_{i^*} = \mathbf{A}\ \mathbf{R}_d^{\mathbf{t}^*[d]} \cdots \mathbf{R}_2^{\mathbf{t}^*[2]}\ \mathbf{R}_1^{\mathbf{t}^*[1]}$. Besides, with the fact that $\mathcal{A}$ either queried the secret key of ID^* after period t^* or never requested it at all, it is clear that $\mathbf{v}$ is not known to $\mathcal{A}$. In this sense, because $\mathbf{v}$ has large min-entropy given $\mathbf{u}^*$, we argue that $\mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*} \ne \mathbf{v}$ with overwhelming probability. Now let $\mathbf{z} = \mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*} - \mathbf{v} \in \mathbb{Z}_q^m$, it holds that i) $\mathbf{z} \ne \mathbf{0}$; ii) $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \mod q$; iii) $\|\mathbf{z}\| \le \sqrt{m} \cdot \|\mathbf{z}\|_\infty \le \sqrt{m} \cdot (\|\mathbf{v}_{\mathsf{ID}'\|\mathbf{t}^*}\|_\infty + \|\mathbf{v}\|_\infty) \le 2\sqrt{m}\beta_d$. $\mathcal{B}$ finally outputs $\mathbf{z}$, which is a valid solution of the given $\mathsf{SIS}_{n,m,q,2\sqrt{m}\beta_d}$ instance.

We observe that the probability that $\mathcal{B}$ does not abort is at least $\frac{1}{q_G \cdot N \cdot T}$, and conditioned on not aborting, it can solve the $\mathsf{SIS}_{n,m,q,2\sqrt{m}\beta_d}$ problem with probability larger than 1/2.

Detector Resistance. We define a sequence of hybrid games G_i^b for $i \in [0,5]$ and G_6, such that game G_0^b, for $b \in \{0,1\}$, is the original experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR-b}}$. We then prove that any two consecutive games are indistinguishable. *Detector resistance* follows from the fact that game G_6 is independent of the bit b. For consistency, use ID_b to denote the involved user ($\mathsf{ID}_b = \mathsf{ID}^*$ or ID_r for $b = 0$ or 1, respectively).

Game G_0^b: This is exactly the original game $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR-b}}$, where $\mathcal{B}$ relies with random strings for oracle queries of $\mathcal{H}_0$ and $\mathcal{G}$.

Game G_1^b: This game is the same as Game G_0^b with only one modification: at the challenge query $\mathsf{Chal}_b^{\mathsf{DR}}$, we utilize the well-designed simulator in [41], so as to produce a simulated proof $\widetilde{\pi}^*$, which is computationally indistinguishable from the real one due to *zero knowledge* of the underlying system.

Game G_2^b: There is one change in Game G_2^b: for the token embedding step in the challenge query, compute the LWE function of revocation token using a random nonce $\mathbf{s}$ instead of the real value $\mathsf{urt}_{\mathsf{ID}_b\|t^*}$, namely, $\mathbf{w}^* = \mathbf{W}\cdot\mathbf{s}+\mathbf{e}^* \mod q$ where $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$. Recall that the current token $\mathsf{urt}_{\mathsf{ID}_b\|t^*} = \mathbf{Q}_1\cdot\mathsf{vdec}_{n,q-1}(\mathsf{urt}_{\mathsf{ID}_b\|t^*-1})+\mathbf{Q}_2\cdot\mathbf{t}^*$ is statistically close to uniform over $\mathbb{Z}_q^n$. Thus, Game G_2^b and G_1^b are statistically indistinguishable.

Game G_3^b: This game follows Game G_2^b with one difference: sample $\mathbf{w}^*$ uniformly from $\mathbb{Z}_q^m$. Note that in the previous game, $\mathbf{W}$ is uniformly random over $\mathbb{Z}_q^{m\times n}$, so the pair $(\mathbf{W},\mathbf{w}^*)$ is a valid $\mathsf{LWE}_{n,q,\chi}$ instance and its distribution is computationally close to the uniform distribution over $\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^m$. Thus, the two games are computationally indistinguishable.

Game G_4^b: This game conducts the same as that in Game G_3^b, except that it uses matrix $\mathbf{B}' \leftarrow U(\mathbb{Z}_q^{n\times m})$ to encrypt users' identity. From Lemma 2, we know that the original matrix $\mathbf{B}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$. Hence, the two games are statistically indistinguishable.

Game G_5^b: This game encrypts the identity with random samples, namely, it generates ciphertexts $\mathbf{c}_1' = \mathbf{z}_1$ and $\mathbf{c}_2' = \mathbf{z}_2 + \lfloor\frac{q}{2}\rfloor\cdot\mathsf{ID}_b$ where $\mathbf{z}_1 \leftarrow U(\mathbb{Z}_q^m)$, $\mathbf{z}_2 \leftarrow U(\mathbb{Z}_q^\ell)$. Based on the hardness of *decision*-LWE, we have that Game G_5^b and G_4^b are computationally indistinguishable.

Game G_6: This game is the same as Game G_5^b except that it replaces the ciphertexts with random vectors, i.e., $\mathbf{c}_1'' = \mathbf{z}_1'$ and $\mathbf{c}_2'' = \mathbf{z}_2'$ where $\mathbf{z}_1' \leftarrow U(\mathbb{Z}_q^m)$, $\mathbf{z}_2' \leftarrow U(\mathbb{Z}_q^\ell)$. Since users' identity is an unknown random string in the view of $\mathcal{A}$, it is clear that Game G_6 and G_5^b are statistically indistinguishable.

Combine the whole analysis above, we have that

$$\mathsf{G}_0^0 \overset{c}{\approx} \mathsf{G}_1^0 \overset{s}{\approx} \mathsf{G}_2^0 \overset{c}{\approx} \mathsf{G}_3^0 \overset{s}{\approx} \mathsf{G}_4^0 \overset{c}{\approx} \mathsf{G}_5^0 \overset{s}{\approx} \mathsf{G}_6,\ \mathsf{G}_6 \overset{s}{\approx} \mathsf{G}_5^1 \overset{c}{\approx} \mathsf{G}_4^1 \overset{s}{\approx} \mathsf{G}_3^1 \overset{c}{\approx} \mathsf{G}_2^1 \overset{s}{\approx} \mathsf{G}_1^1 \overset{c}{\approx} \mathsf{G}_0^1,$$

it then follows that $|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-0} = 1]| = \mathsf{negl}(\lambda)$. This concludes the proof.

Backward Unlinkability. Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{B\text{-}Unlink-b}}$ is much similar to $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR-b}}$, in the sense that the challenger also picks one out of two users to simulate a handshake with $\mathcal{A}$ twice, except now the arbitrary user is predetermined as ID_1. Therefore we can also build a sequence of hybrid games to prove this property as the above constructions, with the only difference that we need to additionally argue the anonymity of revoked users (attribute "backward"). To this effect, it suffices to prove that the publicity of revocation tokens at period t' brings no advantage for $\mathcal{A}$ at period t holding $t < t'$. We tackle this issue in two steps:

First we demonstrate that the update algorithm for revocation token is one-way, i.e., it is impossible to recover a previous token from the current one, the claimed fact is as follows.

Lemma 6. *The update function of revocation token defined in algorithm* $\mathsf{Update}_{\mathsf{U}}$ *is one-way, assuming the hardness of* $\mathsf{ISIS}_{n,q,\sqrt{nk}}$ *problem.*

Proof. Let $\mathbf{u} = \mathsf{urt}_{i\|t} - \mathbf{Q}_2 \cdot \mathbf{t} \in \mathbb{Z}_q^n$, if one can recover the previous token $\mathsf{urt}_{i\|t-1} := \mathbf{v} \in \mathbb{Z}_q^n$ from the current one, satisfying that $\mathsf{urt}_{i\|t} = \mathbf{Q}_1 \cdot \mathsf{vdec}_{n,q-1}(\mathbf{v}) + \mathbf{Q}_2 \cdot \mathbf{t} \mod q$, then one can obtain a non-zero vector $\mathbf{z} = \mathsf{vdec}_{n,q-1}(\mathbf{v}) \in \{0,1\}^{nk}$ such that $\mathbf{Q}_1 \cdot \mathbf{z} = \mathbf{u} \mod q$. In other words, $\mathbf{z}$ is a valid solution to the $\mathsf{ISIS}_{n,q,\sqrt{nk}}$ problem associated with matrix $\mathbf{Q}_1$ and vector $\mathbf{u}$.

Next we show that $\mathcal{A}$ gains no extra advantage after knowing later revocation tokens (e.g., $\mathsf{urt}_{i\|t+1}$). It suffices to prove that $\mathcal{A}$ still can not distinguish the LWE instance $(\mathbf{W}, \mathbf{w}^*)$ in Game G_2^b from real random samples.

Suppose that now Game G_2^b and G_3^b are distinguishable with a non-negligible advantage, which directly implies that $\mathcal{A}$ solves *decision*-LWE with non-negligible probability. It then follows that $\mathcal{A}$ can also solve *search*-LWE with non-negligible probability and a larger sample number $m' = \mathsf{poly}(m)$, implying $\mathcal{A}$ can find the secret token $\mathsf{urt}_{i\|t}$ at current period t by use of $\mathsf{urt}_{i\|t+1}$. In this way, $\mathcal{A}$ will break the one-way property of the update function stated in Lemma 6.

References

1. Abdalla, M., Reyzin, L.: A new forward-secure digital signature scheme. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 116–129. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_10
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_6
3. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller, G.L. (ed.) STOC 1996, pp. 99–108. ACM (1996). https://doi.org/10.1145/237814.237838
4. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011). https://doi.org/10.1007/s00224-010-9278-3
5. An, Z., Zhang, Z., Wen, Y., Zhang, F.: Lattice-based secret handshakes with reusable credentials. In: Gao, D., Li, Q., Guan, X., Liao, X. (eds.) ICICS 2021. LNCS, vol. 12919, pp. 231–248. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88052-1_14
6. Ateniese, G., Kirsch, J., Blanton, M.: Secret handshakes with dynamic and fuzzy matching. In: NDSS 2007. The Internet Society (2007). https://www.ndss-symposium.org/ndss2007/secret-handshakes-dynamic-and-fuzzy-matching/
7. Balfanz, D., Durfee, G., Shankar, N., Smetters, D.K., Staddon, J., Wong, H.: Secret handshakes from pairing-based key agreements. In: S&P 2003, pp. 180–196. IEEE Computer Society (2003). https://doi.org/10.1109/SECPRI.2003.1199336

8. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_28
9. Böhl, F., Hofheinz, D., Jager, T., Koch, J., Striecks, C.: Confined guessing: new signatures from standard assumptions. J. Cryptol. **28**(1), 176–208 (2015). https://doi.org/10.1007/s00145-014-9183-z
10. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_26
11. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) CCS 2004, pp. 168–177. ACM (2004). https://doi.org/10.1145/1030083.1030106
12. Boyen, X., Shacham, H., Shen, E., Waters, B.: Forward-secure signatures with untrusted update. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) CCS 2006, pp. 191–200. ACM (2006). https://doi.org/10.1145/1180405.1180430
13. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988). https://doi.org/10.1016/0022-0000(88)90005-0
14. Brickell, E., Pointcheval, D., Vaudenay, S., Yung, M.: Design validations for discrete logarithm based signature schemes. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 276–292. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-540-46588-1_19
15. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_16
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
17. Castelluccia, C., Jarecki, S., Tsudik, G.: Secret handshakes from CA-oblivious encryption. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 293–307. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_21
18. ETSI: ETSI TR 103 570: CYBER; Quantum-Safe Key Exchange, 1.1.1 edn (2017)
19. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC 2008, pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407
20. Hou, L., Lai, J., Liu, L.: Secret handshakes with dynamic expressive matching policy. In: Liu, J.K., Steinfeld, R. (eds.) ACISP 2016. LNCS, vol. 9722, pp. 461–476. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40253-6_28
21. Itkis, G., Reyzin, L.: Forward-secure signatures with optimal signing and verifying. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 332–354. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_20
22. Jarecki, S., Kim, J., Tsudik, G.: Group secret handshakes or affiliation-hiding authenticated group key agreement. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 287–308. Springer, Heidelberg (2006). https://doi.org/10.1007/11967668_19
23. Jarecki, S., Liu, X.: Private mutual authentication and conditional oblivious transfer. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 90–107. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_6
24. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions.

In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 373–403. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_13
25. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 101–131. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_4
26. Libert, B., Yung, M.: Dynamic fully forward-secure group signatures. In: Feng, D., Basin, D.A., Liu, P. (eds.) ASIACCS 2010, pp. 70–81. ACM (2010). https://doi.org/10.1145/1755688.1755698
27. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 107–124. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_8
28. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Forward-secure group signatures from lattices. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. LNCS, vol. 11505, pp. 44–64. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_3
29. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007). https://doi.org/10.1137/S0097539705447360
30. Michalevsky, Y., Nath, S., Liu, J.: Mashable: mobile applications of secret handshakes over bluetooth LE. In: Chen, Y., Gruteser, M., Hu, Y.C., Sundaresan, K. (eds.) MobiCom 2016, pp. 387–400. ACM (2016). https://doi.org/10.1145/2973750.2973778
31. Nakanishi, T., Funabiki, N.: Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 533–548. Springer, Heidelberg (2005). https://doi.org/10.1007/11593447_29
32. Nakanishi, T., Hira, Y., Funabiki, N.: Forward-secure group signatures from pairings. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 171–186. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03298-1_12
33. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) STOC 2009, pp. 333–342. ACM (2009). https://doi.org/10.1145/1536414.1536461
34. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
35. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC 2005, pp. 84–93. ACM (2005). https://doi.org/10.1145/1060590.1060603
36. Song, D.X.: Practical forward secure group signature schemes. In: Reiter, M.K., Samarati, P. (eds.) CCS 2001, pp. 225–234. ACM (2001). https://doi.org/10.1145/501983.502015
37. Tian, Y., Li, Y., Zhang, Y., Li, N., Yang, G., Yu, Y.: DSH: deniable secret handshake framework. In: Su, C., Kikuchi, H. (eds.) ISPEC 2018. LNCS, vol. 11125, pp. 341–353. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99807-7_21
38. Wen, Y., Zhang, F.: A new revocable secret handshake scheme with backward unlinkability. In: Camenisch, J., Lambrinoudakis, C. (eds.) EuroPKI 2010. LNCS, vol. 6711, pp. 17–30. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22633-5_2

39. Wen, Y., Zhang, F., Wang, H., Gong, Z., Miao, Y., Deng, Y.: A new secret handshake scheme with multi-symptom intersection for mobile healthcare social networks. Inf. Sci. **520**, 142–154 (2020)
40. Xu, S., Yung, M.: k-anonymous secret handshakes with reusable credentials. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) CCS 2004, pp. 158–167. ACM (2004). https://doi.org/10.1145/1030083.1030105
41. Yang, R., Au, M.H., Zhang, Z., Xu, Q., Yu, Z., Whyte, W.: Efficient lattice-based zero-knowledge arguments with standard soundness: construction and applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 147–175. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_6
42. Zhang, Z., Zhang, F., Tian, H.: CSH: a post-quantum secret handshake scheme from coding theory. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12309, pp. 317–335. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59013-0_16
43. Zhou, L., Susilo, W., Mu, Y.: Three-round secret handshakes based on ElGamal and DSA. In: Chen, K., Deng, R., Lai, X., Zhou, J. (eds.) ISPEC 2006. LNCS, vol. 3903, pp. 332–342. Springer, Heidelberg (2006). https://doi.org/10.1007/11689522_31