# Lattice-Based Secret Handshakes with Reusable Credentials

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

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
{anzhy,zhangzhr26}@mail2.sysu.edu.cn, isszhfg@mail.sysu.edu.cn
[2] Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China
[3] School of Statistics and Mathematics, Guangdong University of Finance and Economics, Guangzhou 510320, China
wenyamin@gdufe.edu.cn

**Abstract.** Secret handshake, as a fundamental privacy-preserving primitive, allows members in the same organization to anonymously authenticate each other. Since its proposal in 2003, numerous schemes have been presented in terms of various security, efficiency, and functionality. Unfortunately, all of the contemporary designs are based on number theoretic assumptions and will be fragile in the setting of quantum computations. In this paper, we fill this gap by presenting the first lattice-based secret handshake scheme with reusable credentials. More precisely, we utilize the verifier-local revocation techniques for member secession, such that users' credentials support reusability rather than one-time usage. To build an interactive authentication protocol, we subtly modify a Stern-type zero-knowledge argument by use of a key exchange protocol, which enables users to negotiate a session key for further communication. The security of our scheme relies on the Short Integer Solution (SIS) and Learning With Errors (LWE) assumptions.

**Keywords:** Secret handshake · Lattice cryptography · Zero-knowledge · Privacy-preserving · Mutual authentication

## 1 Introduction

Secret Handshake Scheme, firstly introduced by Balfanz *et al.* [5], is designed for realizing mutually anonymous authentication. In secret handshakes, potential users form different groups and one will reveal his/her affiliation to another if and only if both of them belong to the same organization. Thus the interactive protocol run between users from different groups will leak nothing about their identities and affiliations. Moreover, members keep responsible for the handshakes they execute since a tracing algorithm will identify them should the need occurs. Following the initial work in [5], many secret handshake schemes have been proposed based on different cryptography techniques. Some of them used one-time


D. Gao et al. (Eds.): ICICS 2021, LNCS 12919, pp. 231–248, 2021.
https://doi.org/10.1007/978-3-030-88052-1_14

pseudonyms in their constructions [7,14,26,29,33]. Whereas, a more efficient design for unlinkability is to use reusable credentials. For better efficiency, Xu and Yung [30] presented the first such scheme with somewhat weaker unlinkability. Ateniese *et al.* [4] proposed an improved unlinkable secret handshake scheme secure in the standard model. Subsequently, Jarecki and Liu [15] proposed a practical unlinkable secret handshake scheme achieving both traceability and revocation with reusable certificates. From then on, many unlinkable secret handshake schemes achieving more requirements were proposed [13,17,25,28]. Some practical applications of secret handshakes in social networks were also exploited, such as online dating, anonymous services of e-commerce and e-healthcare [12].

Since the integrated systems offering authentication interface always maintain high staff turnover, one desirable functionality of secret handshake is the support for membership revocation, i.e., users can leave or be revoked from the group. Early attempts to capture this property need the whole system to be re-initialized (including group public keys and users' secret keys), which obviously bring unsuitable workloads to all involved parties. Another flexible approach, verifier-local revocation (VLR), is formalized by Boneh and Shacham [6] and allows revoking a group member in a simpler manner. It only requires the corresponding verifiers to download an updatable revocation list. Jarecki and Liu [15] first employed a VLR group signature to design a secret handshake. Although their construction shows a heuristic relation between the VLR group signature and secret handshake, the aforementioned scheme employed an additional technique, i.e., a private Conditional Oblivious Transfer for relations on discrete logarithm representations. Besides, as pointed out in [27], their scheme only provided a generic construction and may be too complicated to be implemented. The above unsatisfactory situation encourages us to design a more compact scheme with flexible user management.

In addition, nearly all the known secret handshake schemes are designed on the hardness of factoring integers or the discrete logarithm problem. These constructions will be insecure once quantum computers become a reality. To our best knowledge, the only known post-quantum secret handshake scheme was proposed by Zhang *et al.* [32] using one-time pseudonyms from coding theory. However, we observe that due to improper adaptation of Stern's identification system, challenges used in their scheme are independent of the commitments generated from user's secrets. Therefore an adversary, who has no valid group credential, can always utilize simulated zero-knowledge proof to forge an authentication code, so as to conduct a successful handshake. As for other post-quantum candidates, lattice-based cryptography is considered to be very promising and enjoys provable security under worst-case hardness assumptions. Further, we observe that group signature, another privacy-preserving primitive analogous to secret handshake, has made some inspiring breakthroughs in lattice theory [18,19,21]. Thus, it is worthwhile to explore the area of lattice-based secret handshakes. To fill this deficiency, we may need some adaptive and insightful ideas.

Our Contributions and Techniques. Inspired by the VLR group signature [18], we introduce the first lattice-based secret handshake scheme with reusable credentials.

Consider the system having $N = 2^\ell$ members for each group, user's identity $\mathsf{ID}$ is represented by a binary index $d \in \{0,1\}^\ell$. To generate a reusable group credential, the identity is embedded into the user's secret key $\mathsf{usk} := \mathbf{x} \in \mathbb{Z}^{(2\ell+1)m}$ by setting $\mathbf{x}_k^{d[k]} := \mathbf{0}^m$ for $k \in [\ell]$. Indeed, $\mathbf{x}$ is a $\beta$-bounded solution to the $\mathsf{ISIS}$ instance determined by the Bonsai signature, holding that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u}$, where $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\ldots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1|]$ specifies the structure of a Bonsai tree. Furthermore, user's revocation token ($\mathsf{urt}$) is constructed via the root element of $\mathbf{A}$ and $\mathbf{x}$, i.e., $\mathsf{urt} := \mathbf{A}_0 \cdot \mathbf{x}_0$, according to VLR feature. The above realization guarantees the secrecy of user's credential, such that it can be reusable instead of one-time usage for one handshake.

The major difficulty we have overcome lies in how to modify the zero-knowledge argument to fit our handshake protocol. Generally, a Stern-like zero-knowledge argument system has three components: three commitments $cmt = (c_1, c_2, c_3)$, a challenge $ch = \mathcal{H}_0(cmt, \cdot) \in \{1,2,3\}$ and a response $rsp$, which is used to recover and verify 2 of 3 commitments according to the value of $ch$ (e.g., check $(c_2, c_3)$ for $ch = 1$). Since a secret handshake scheme is a mutual anonymous authentication protocol, we need to cut off the functionality of directly verifying the generated argument for the receiver. Therefore, instead of sending $cmt$, we dispatch a partial commitment value $\overline{cmt}$, consisting of $1/3$ of commitments that can not be checked by the corresponding response (e.g., $\overline{cmt} = c_1$ for $ch = 1$). Next, we change the challenge $ch$ as $ch := ch \oplus \mathbf{m}$, where $\mathbf{m}$ is a hidden message utilized to conduct an $\mathsf{LWE}$-based key exchange [8]. After the above adjustment, both participants can first calculate the reserved $2/3$ of commitments[1] via received responses $rsp$, and then recover the original $cmt$ combining $\overline{cmt}$. Further, they can retrieve the hidden message $\mathbf{m} := ch \oplus \mathcal{H}_0(cmt, \cdot)$ to produce a session key $\mathbf{K}$. In the end, a message authentication code $\mathsf{V} = \mathcal{H}_2(\mathbf{K}\|\mathbf{m}, \cdot)$ is used to determine the result (0 or 1) of a handshake. In this way, we also fix the flaw of Zhang *et al.*'s scheme [32]. We elaborate more details on this strategy in algorithm $\mathsf{Handshake}$ of our scheme.

To summarize, by employing the setting of VLR, our scheme supports reusable credentials and it only requires active users to download the published revocation token list for group updates. The whole revocation tokens will serve as a tracing secret key kept by group authority. Besides, we mask a secret message with a modified Stern-type zero-knowledge argument [18], which ensures that the interactive handshake protocol can negotiate a session key for both participants and also prevents the attack of detection.

Organization. The remainder of this paper is organized as follows. In Sect. 2, we recall our preliminaries including some lattice techniques and the underlying argument system. Model and security requirements of secret handshakes are reviewed in Sect. 3. In Sect. 4, we describe our secret handshake scheme. The security and performance analysis are depicted in Sect. 5.

[1] Note that they can not verify these commitments since they do not have the original ones.

## 2 Preliminaries

*Notations.* Vectors will be denoted in bold lower-case letters and matrices will be denoted in bold upper-case letters. We assume that all vectors are column vectors. Let $\|\cdot\|$ and $\|\cdot\|_\infty$ denote the Euclidean norm ($\ell_2$) and infinity norm ($\ell_\infty$) of a vector respectively. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ is denoted by $(\mathbf{x}\|\mathbf{y})$, and 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 a positive integer $n$, let $[n]$ denote the set $\{1, \ldots, n\}$. If $\mathsf{S}$ is a finite set, $y \xleftarrow{\$} \mathsf{S}$ means that $y$ is chosen uniformly at random from $\mathsf{S}$. For $a \in \mathbb{R}$, use $\log a$ and $\exp(a)$ to denote the logarithm and the power of $a$ with base 2 and $e$, respectively.

### 2.1 Background on Lattices

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 $\Lambda^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\cdot\mathbf{x} = \mathbf{u} \mod q\}$.

**Gaussians over Lattices.** For any positive real $\sigma$ and $n$-dimensional lattice $\Lambda$, the $n$-dimensional Gaussian function and the discrete Gaussian distribution over $\Lambda$ are defined as: $\forall \mathbf{x} \in \mathbb{R}^n, \rho_\sigma(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$; $\forall \mathbf{x} \in \Lambda, D_{\Lambda,\rho}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda)}$.

**Lemma 1** ([9]). *Let $n$ and $q \geq 2$ be integers. Let $m \geq 2n\log q$, and $\sigma \geq \omega(\sqrt{\log m})$.*

*1. For all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, for $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u} = \mathbf{A}\cdot\mathbf{x} \mod q$ is statistically close to uniform over $\mathbb{Z}_q^n$.*
*2. For $\beta = \lceil\sigma\cdot\log m\rceil$, and $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, $\Pr[\|\mathbf{x}\|_\infty > \beta]$ is negligible.*

**Computational Lattice Problems.** The following are the definitions and hardness results of $\mathsf{SIS}$, $\mathsf{ISIS}$ ($\ell_\infty$ norm) and $\mathsf{LWE}$, which will be used in this work.

**Definition 1** ([1,9]). *The $\mathsf{SIS}^\infty_{n,m,q,\beta}$ and $\mathsf{ISIS}^\infty_{n,m,q,\beta}$ with parameters $(n, m, q, \beta)$ are as follows: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$,*

- $\mathsf{SIS}^\infty_{n,m,q,\beta}$*: to find a non-zero vector $\mathbf{x} \in \Lambda^\perp(\mathbf{A})$ such that $\|\mathbf{x}\|_\infty \leq \beta$.*
- $\mathsf{ISIS}^\infty_{n,m,q,\beta}$*: to find a vector $\mathbf{x} \in \Lambda^{\mathbf{u}}(\mathbf{A})$ such that $\|\mathbf{x}\|_\infty \leq \beta$.*

The hardness of the $\mathsf{SIS}$ and $\mathsf{ISIS}$ problems is given by a worst-case to average-case reduction from standard lattice problems, such as $\mathsf{SIVP}$.

**Definition 2** ([23]). *Let $n, m \geq 1, q \geq 2$, and let $\chi$ 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 $(\mathbf{a}, \mathbf{a}^\top\cdot\mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The $\mathsf{LWE}_{n,q,\chi}$ problem is to distinguish $m$ samples from $A_{\mathbf{s},\chi}$ (let $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and $m$ samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

If $q$ is a prime power, $\beta \geq \sqrt{n}\mathcal{O}(\log n), \gamma = \widetilde{\mathcal{O}}(nq/\beta)$, and $\chi$ is a $\beta$-bounded distribution (i.e., $\chi = D_{\mathbb{Z}_m,\sigma}$), $\mathsf{LWE}_{n,q,\chi}$ problem is as least as hard as $\mathsf{SIVP}_\gamma$.

**Lattice Algorithms.** The following facts describe two fundamental tools in lattice-based cryptography: the trapdoor generation and the preimage sampling algorithms. We use them to generate group/user keys in our scheme.

**Lemma 2 ([2,3,20]).** *Given integers $n \geq 1$, $q \geq 2$, and $m \geq 2n\log q$. There is a* PPT *algorithm* $\mathsf{GenTrap}(n, m, q)$ *that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{R}_\mathbf{A}$, such that $\mathbf{A}$ is statistically close to uniform in $\mathbb{Z}_q^{n\times m}$ and $\mathbf{R}_\mathbf{A}$ is a basis for $\Lambda^\perp(\mathbf{A})$. Moreover, for any vector $\mathbf{u} \in \mathbb{Z}_q^n$ and $\sigma = \omega(\sqrt{n\log q \log n})$, there is a* PPT *algorithm* $\mathsf{SamplePre}(\mathbf{R}_\mathbf{A}, \mathbf{A}, \mathbf{u}, \sigma)$ *that outputs $\mathbf{x} \in \Lambda^\mathbf{u}(\mathbf{A})$ from a distribution that is with negligible distance from $D_{\Lambda^\mathbf{u}(\mathbf{A}),\sigma}$.*

### 2.2 Zero-Knowledge Arguments of Knowledge

In a zero-knowledge argument of knowledge ($\mathsf{ZKAoK}$) system, a prover proves his/her possession of some witness for an NP relation to a verifier, without revealing any additional information. Generally, a secure $\mathsf{ZKAoK}$ must satisfy three requirements: *completeness, proof of knowledge* and *zero knowledge* [10].

In [18], Langlois et al. proposed a Stern-type $\mathsf{ZKAoK}$ over lattices for the following relation:

$$\begin{cases} d = d[1]\dots d[\ell] \in \{0,1\}^\ell,\ \mathbf{e} \in \mathbb{Z}^m; \\ \mathbf{x}_k^{1-d[k]} = \mathbf{0}^m,\ \forall k \in [\ell],\ \mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}; \\ \mathbf{A}\cdot\mathbf{x} = \mathbf{u} \mod q,\ \|\mathbf{x}\|_\infty \leq \beta; \\ \mathbf{W}\cdot(\mathbf{A}_0\cdot\mathbf{x}_0) + \mathbf{e} = \mathbf{w} \mod q,\ \|\mathbf{e}\|_\infty \leq \beta, \end{cases} \tag{1}$$

where the tuple $(d, \mathbf{x}, \mathbf{e})$ is the secret witness and $(\mathbf{A}, \mathbf{W}, \mathbf{u}, \mathbf{w})$ is the public input. The above protocol has perfect completeness, soundness error $2/3$ with a statistical simulator, and an efficient knowledge extractor. Further, by use of Fiat-Shamir heuristic, it can be transformed into an $\mathsf{NIZKAoK}$ termed as a triple

$$\Pi = (\{cmt^k\}_{k=1}^t, \{ch^k\}_{k=1}^t, \{rsp^k\}_{k=1}^t), \tag{2}$$

where $cmt^k = \langle cmt^k(1), cmt^k(2), cmt^k(3)\rangle$ for $k \in [t]$ and $\{ch^k\}_{k=1}^t = \mathcal{H}_0(\mathbf{A}, \mathbf{W}, \mathbf{u}, \mathbf{w}, \{cmt^k\}_{k=1}^t) \in \{1,2,3\}^t$. We utilize this protocol as an underlying building block and refer readers to [18, Sec. 4] for a more detailed description. Security of the aforementioned $\mathsf{ZKAoK}$ is under the hardness assumption of $\mathsf{SIS}$.

## 3 Model and Security Properties of Secret Handshake

In this section, we review the model and security definitions for a secret handshake scheme (SHS). An SHS involves several entities: a group authority $\mathsf{GA}$ that manages members' enrollment and revocation, as well as tracing users' malicious behaviors, and s set of users who are potential group members. Based on the previous definitions in [5,7], an SHS consists of the following algorithms:

- Setup: On input security parameter $\lambda$, this algorithm generates the public parameters par, which is common to all subsequently established groups.
- CreateGroup: It is a key generation algorithm executed by GA to create a group $G$. On input par, this algorithm outputs group public key and secret key $(\mathsf{gpk}, \mathsf{gsk})$.
- AddMember: It is a two-party algorithm run by GA, which certifies a user to become a legitimate group member. After verifying the user's real identity, GA issues the user's group credential Cred (including group identity ID).
- Handshake: This algorithm is a mutual authentication protocol between two active members $(A, B)$. It outputs 1 and produces a session key for both parties if and only if $A$ and $B$ belong to the same group.
- TraceMember: It is a polynomial time algorithm executed by GA. When a transcript T of a secret handshake between user $A$ and $B$ is submitted, GA outputs the identities of user $A$ and $B$ via secret key gsk.
- RemoveMember: It is a polynomial time algorithm authorized by GA. Taking the current credential revocation list ($\mathcal{CRL}$) and the target user's credential as input, it outputs an up-to-date list $\mathcal{CRL}$ to revoke an active member.

As considered in [4,5], an SHS must satisfy some security requirements: *completeness*, *impersonator resistance*, *detector resistance*, *unlinkability*. They are stated via the corresponding experiments below, respectively. Use CoU and CoG to denote the corruption list of users and groups, respectively. The involved oracles are listed as follows:

KeyP(par): this oracle simulates to create a new group and returns gpk to $\mathcal{A}$.
AddM$(U, G)$: this oracle adds a puppet user $U$ to the chosen group $G$. Then it returns the user's credential Cred to $\mathcal{A}$ and adds ID to corruption list Cor, which is initialized as $\emptyset$.
CorU(ID, G): this oracle returns user's Cred whose identity in group $G$ is ID to $\mathcal{A}$, then it adds (ID, G) to list CoU.
KeyG(par): this oracle returns secret key gsk of some group $G$ and adds $G$ to CoG, implying that $G$ is under the control of $\mathcal{A}$.
HS(ID): this oracle simulates a two-party handshake by generating the interactive transcripts. In particular, the adversary can request the hash functions and valid NIZKAoK used in algorithm Handshake on any random witness.
Trace(T): this oracle returns the identities of users involved in the handshake transcript T. Note that this oracle is only allowed to be queried for transcripts that are not generated from the game between $\mathcal{A}$ and the challenger.

**Completeness** makes sure that the secret handshake protocol always outputs 1 when the interactive participants belong to the same group, and that algorithm TraceMember can always identify the involved users.

**Impersonator resistance** demands that an adversary, who attempts to impersonate a legitimate member of an uncorrupted group, can only succeed with a negligible probability.

**Definition 3.** *Impersonator resistance is achieved if, for any* PPT *adversary, the following experiment returns 1 with negligible probability.*

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

---

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, $\mathsf{CoG}$, $\mathsf{CoU} := \emptyset$.
$(\mathsf{gpk}) \leftarrow \mathcal{A}^{\mathsf{KeyP}}(\mathsf{par})$.
Return 0 if $\mathsf{gpk}$ is not well-formed.
$(\mathsf{ID}^*, G^*) \leftarrow \mathcal{A}^{\mathsf{AddM},\mathsf{CorU},\mathsf{KeyG},\mathsf{HS},\mathsf{Trace}}(\mathsf{gpk})$.
Return 1 if $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}^*) = 1 \wedge G^* \notin \mathsf{CoG} \wedge (\cdot, G^*) \notin \mathsf{CoU}$.

**Detector resistance** requires that an adversary will only succeed with a negligible probability when he activates a handshake protocol with an honest user to identify his/her affiliation. Namely, it's infeasible to detect a user's affiliation without the corresponding group secret key.

**Definition 4.** *Detector resistance is achieved if, for any* PPT *adversary, the absolute difference of probability of outputting 1 between experiment* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-1}$ *and* $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-0}$ *is negligible.*

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

---

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, $\mathsf{CoG}$, $\mathsf{CoU} := \emptyset$.
$(\mathsf{gpk}) \leftarrow \mathcal{A}^{\mathsf{KeyP}}(\mathsf{par})$.
Return 0 if $\mathsf{gpk}$ is not well-formed.
$(\mathsf{ID}^*, G^*) \leftarrow \mathcal{A}^{\mathsf{AddM},\mathsf{CorU},\mathsf{KeyG},\mathsf{HS},\mathsf{Trace}}(\mathsf{gpk})$, holding $G^* \notin \mathsf{CoG} \wedge (\cdot, G^*) \notin \mathsf{CoU}$.
if $b = 0$ : $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}^*)$*;*
if $b = 1$ : $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}_r)$. $\mathsf{ID}_r$ is an arbitrary active user (not $\mathsf{ID}^*$).
$b^* \leftarrow \mathcal{A}^{\mathsf{AddM},\mathsf{CorU}(\neg\{\mathsf{ID}^*,\mathsf{ID}_r\}),\mathsf{KeyG}(\neg\{G^*,G_r\}),\mathsf{HS},\mathsf{Trace}}(\mathsf{gpk})$.
Return 1 if $b^* = b$ else return 0.

**Unlinkability** ensures that no adversary can distinguish whether two executions of secret handshake protocol involve the same honest and active user with a non-negligible probability.

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

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

---

$\mathsf{par} \leftarrow \mathsf{Setup}(\lambda)$, $\mathsf{CoG}$, $\mathsf{CoU} := \emptyset$.
$(\mathsf{gpk}) \leftarrow \mathcal{A}^{\mathsf{KeyP}}(\mathsf{par})$.
Return 0 if $\mathsf{gpk}$ is not well-formed.
$(\mathsf{ID}_0, G_0, \mathsf{ID}_1, G_1) \leftarrow \mathcal{A}^{\mathsf{AddM},\mathsf{CorU},\mathsf{KeyG},\mathsf{HS},\mathsf{Trace}}(\mathsf{gpk})$,
holding that $G_i \notin \mathsf{CoG} \wedge (\mathsf{ID}_i, G_i) \notin \mathsf{Cor} \cup \mathcal{CRL}$ for $i \in \{0, 1\}$.
if $b = 0$ : $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}_0)$, $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}_0)$*;*
if $b = 1$ : $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}_0)$, $\mathsf{Handshake}(\mathcal{A}, \mathsf{ID}_1)$.

$b^* \leftarrow \mathcal{A}^{\mathsf{AddM},\mathsf{CorU}(\neg\{\mathsf{ID}_0,\mathsf{ID}_1\}),\mathsf{KeyG}(\neg\{G_0,G_1\}),\mathsf{HS},\mathsf{Trace}}(\mathsf{gpk})$.
Return 1 if $b^* = b$ else return 0.

## 4 Our Lattice-Based Secret Handshake Scheme

In this section, we describe how to, relying on the technique of VLR, modify and apply the Stern-like ZKAoK [18] to construct a lattice-based SHS with reusable credentials, which satisfies the security requirements in Sect. 3. As the setting in [18], we assume that the group of our scheme has a maximum number of members $N$. Procedures for building our scheme are depicted as follows.

- Setup: Given a security parameter $\lambda$, this algorithm specifies the following:
  - A maximum number of group members $N = 2^\ell = \mathsf{poly}(\lambda)$.
  - Dimension $n = \mathcal{O}(\lambda)$, prime modulus $q = \omega(n^2 \log n)$ and matrix dimension $m \geq 2n \log q$.
  - Matrix dimensions $m_1 = \mathsf{poly}(n)$, integer modulus $q_1 \leq 2^{\mathsf{poly}(n)}$, and an integer $\theta \geq 2\lambda/(nm_1)$ for the session key exchange.
  - Gaussian parameter $\sigma = \omega(\sqrt{n \log q \log n})$ and integer norm bound $\beta = \lceil \sigma \cdot \log m \rceil$.
  - A $\beta$-bounded distribution $\chi = D_{\mathbb{Z}_m,\sigma}$ for the LWE function.
  - Discrete Gaussian distribution $\chi_1$ over $\mathbb{Z}$ with standard deviation $\sigma_1 > \sqrt{2n/\pi}$.
  - A random matrix $\mathbf{K} \in \mathbb{Z}_{q_1}^{n \times m_1}$.
  - An injective map $F : \mathbb{Z}_{q_1}^{n \times m_1} \rightarrow \{1,2,3\}^t$, where $t = \omega(\log n)$ is the number of argument repetitions. $F^{-1}$ is the inverse of $F$.
  - Two random oracles: $\mathcal{H}_0 : \{0,1\}^* \rightarrow \{1,2,3\}^t$ and $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^{m \times n}$. A secure hash function $\mathcal{H}_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^n$.

The algorithm outputs global public parameters

$$\mathsf{par} = (N, \ell, n, q, m, m_1, q_1, \theta, \sigma, \beta, \chi, \chi_1, \sigma_1, \mathbf{K}, F, F^{-1}, t, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2).$$

- CreateGroup: GA takes par as input to create a group $G$. GA works as follows:
  - Run $\mathsf{GenTrap}(n, m, q)$ to get $\mathbf{A}_0 \in \mathbb{Z}^{n \times m}$ and trapdoor $\mathbf{R}$.
  - Sample $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}^{n \times m}$ for all $b \in \{0,1\}$ and $i \in [\ell]$. Then define the matrix

$$\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \ldots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n \times (2\ell+1)m}. \tag{3}$$

  - For group user with index $d \in [N]$, let $d[1] \ldots d[l] \in \{0,1\}^\ell$ denote the binary representation of $d$, and do the following:
    - Sample vectors $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, and then compute $\mathbf{z} = \sum_{i=1}^{\ell-1} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]} \mod q$. Run $\mathsf{SamplePre}(\mathbf{R}, \mathbf{A_0}, \mathbf{u} - \mathbf{z}, \sigma)$ to get $\mathbf{x}_0 \in \mathbb{Z}^m$. Let $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ be zero-vectors $\mathbf{0}^m$, and define $\mathbf{x}^{(d)} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$. If $\|\mathbf{x}^{(d)}\|_\infty > \beta$ with negligible probability then repeat this step.

– Set the user secret key as $\mathsf{usk}[d] = \mathbf{x}^{(d)}$, and the revocation token as $\mathsf{urt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \in \mathbb{Z}_q^n$.

Finally GA sets the group public and secret key as $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, $\mathsf{gsk} = (\mathbf{R}, \{\mathsf{usk}[d], \mathsf{urt}[d]\}_{d=1}^N)$, respectively. Then GA builds users' identities $\mathsf{ID} = \{d\}_{d=1}^N$, revocation list $\mathcal{CRL} = \{\emptyset\}$ and member list $\mathcal{L} = \{\emptyset\}$.

– AddMember: When a user $U$ wants to join the group $G$, GA chooses a spare $d_u$ as user's $\mathsf{ID}_u$ and issues $U$'s credential as $\mathsf{Cred}_u = (\mathsf{ID}_u, \mathsf{usk}[d_u], \mathsf{urt}[d_u])$. Then GA sends $\mathsf{Cred}_u$ to the user and adds $(U, \mathsf{ID}_u)$ to $\mathcal{L}$.
– Handshake: Suppose a member $A$ from group $G_1$ with $\mathsf{gpk}_1 = (\mathbf{A}, \mathbf{u}_1)$, $\mathsf{Cred}_a = (d_a, \mathsf{usk}_a, \mathsf{urt}_a)$, credential revocation list $\mathcal{CRL}_1$, and another member $B$ from group $G_2$ with $\mathsf{gpk}_2 = (\mathbf{B}, \mathbf{u}_2)$, $\mathsf{Cred}_b = (d_b, \mathsf{usk}_b, \mathsf{urt}_b)$, credential revocation list $\mathcal{CRL}_2$, engage in a handshake protocol.
  1. $A \to B : (\mathtt{PROOF}_a)$
     (a) $A$ samples a private key $\mathbf{S}_a \hookleftarrow \chi(\mathbb{Z}_{q_1}^{n_3 \times m_1})$ and a small noise $\mathbf{E}_a \hookleftarrow \chi(\mathbb{Z}_{q_1}^{n_3 \times m_1})$. Then $A$ computes $\mathbf{C}_a = \mathbf{K} \cdot \mathbf{S}_a + \mathbf{E}_a \in \mathbb{Z}_{q_1}^{n_3 \times m_1}$.
     (b) $A$ samples $\mathbf{e}_a \hookleftarrow \chi^m$ and $\rho_a \xleftarrow{\$} \{0,1\}^n$. Then $A$ computes $\mathbf{W}_a = \mathcal{H}_1(\mathbf{A}, \mathbf{u}_1, \mathbf{K}, \rho_a) \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{w}_a = \mathbf{W}_a \cdot \mathsf{urt}_a + \mathbf{e}_a \mod q$.
     (c) $A$ repeats $t$ times the underlying ZKAoK protocol [18, Sec. 4.1] with public parameter $(\mathbf{A}, \mathbf{u}_1, \mathbf{W}_a, \mathbf{w}_a)$ and witness $(d_a, \mathsf{usk}_a, \mathbf{e}_a)$, then makes it non-interactive with the Fiat-Shamir heuristic as a triple $\mathit{\Pi}_a = (\{cmt_a^k\}_{k=1}^t, ch_a, \{rsp_a^k\}_{k=1}^t)$, where

$$ch_a = (\{ch_a^k\}_{k=1}^t) = \mathcal{H}_0(\mathbf{A}, \mathbf{u}_1, \mathbf{W}_a, \mathbf{w}_a, \{cmt_a^k\}_{k=1}^t) \oplus F(\mathbf{C}_a). \quad (4)$$

     (d) Denote the commitment values which will not be checked as $\overline{cmt}_a = (\overline{cmt}_a^1, \ldots, \overline{cmt}_a^t)$. Namely, $A$ sets

$$\overline{cmt}_a^k = \begin{cases} \langle cmt_a^k(1) \rangle, & ch_a^k = 1; \\ \langle cmt_a^k(2) \rangle, & ch_a^k = 2; \\ \langle cmt_a^k(3) \rangle, & ch_a^k = 3. \end{cases} \quad (5)$$

     (e) $A$ sets $\mathtt{PROOF}_a = (\overline{cmt}_a, ch_a, \{rsp_a^k\}_{k=1}^t, \rho_a, \mathbf{w}_a)$ and sends it to $B$.
  2. $B \to A : (\mathtt{PROOF}_b, \mathtt{V}_b)$
     (a) $B$ sets $\mathbf{W}'_a = \mathcal{H}_1(\mathbf{B}, \mathbf{u}_2, \mathbf{K}, \rho_a)$. Then for each $\mathbf{v}_i \in \mathcal{CRL}_2$, $B$ computes $\mathbf{e}'_i = \mathbf{w}_a - \mathbf{W}'_a \cdot \mathbf{v}_i$. If there exists an index $i$ such that $\|\mathbf{e}'_i\|_\infty \le \beta$, $B$ sends $A$ a random pair $(\mathtt{PROOF}_b, \mathtt{V}_b)$ and aborts.
     (b) $B$ samples his ephemeral key $\mathbf{S}_b \hookleftarrow \chi(\mathbb{Z}_{q_1}^{m_1 \times n_3})$ and a small noise $\mathbf{E}_b \hookleftarrow \chi(\mathbb{Z}_{q_1}^{m_1 \times n_3})$. Then $B$ computes $\mathbf{C}_b = \mathbf{K} \cdot \mathbf{S}_b + \mathbf{E}_b \in \mathbb{Z}_{q_1}^{m_1 \times n_3}$.
     (c) $B$ computes the checked value $cmt_a^* = (cmt_a^{*1}, \ldots, cmt_a^{*t})$ for $k \in [t]$ from corresponding $rsp_a^k$ and $ch_a^k$. Namely, $B$ computes

$$cmt_a^{*k} = \begin{cases} \langle cmt_a^{*k}(2), cmt_a^{*k}(3) \rangle, & ch_a^k = 1; \\ \langle cmt_a^{*k}(1), cmt_a^{*k}(3) \rangle, & ch_a^k = 2; \\ \langle cmt_a^{*k}(1), cmt_a^{*k}(2) \rangle, & ch_a^k = 3. \end{cases} \quad (6)$$

     Details of the above calculations are depicted in [18, Sec. 4.1].

(d) By proper concatenations and rearrangements of $cmt_a^*$ and $\overline{cmt}_a$ according to $ch_a$, $B$ recovers the original commitments $cmt'_a$ (e.g., set $cmt'^k_a = (cmt^{*k}_a(2), \overline{cmt}^k_a, cmt^{*k}_a(3))$ if $ch^k_a = 2$). Then $B$ retrieves the hidden message $\mathbf{C}'_a = F^{-1}(ch_a \oplus \mathcal{H}_0(\mathbf{B}, \mathbf{u}_2, \mathbf{W}'_a, \mathbf{w}_a, cmt'_a))$.

(e) $B$ computes $\mathbf{w}_b = \mathbf{W}_b \cdot \mathsf{urt}[d_b] + \mathbf{e}_b \mod q$, where $\rho_b \xleftarrow{\$} \{0,1\}^n$, $\mathbf{W}_b = \mathcal{H}_1(\mathbf{B}, \mathbf{u}_2, \mathbf{K}, \rho_b)$ and $\mathbf{e}_b \hookleftarrow \chi^m$.

(f) Similarly with public input $(\mathbf{B}, \mathbf{u}_2, \mathbf{W}_b, \mathbf{w}_b)$ and witness $(d_b, \mathsf{usk}[d_b], \mathbf{e}_b)$, $B$ runs the underlying $\mathsf{ZKAoK}$ to get a triple $\Pi_b = (\{cmt^k_b\}^t_{k=1}, ch_b, \{rsp^k_b\}^t_{k=1})$, where

$$ch_b = (\{ch^k_b\}^t_{k=1}) = \mathcal{H}_0(\mathbf{B}, \mathbf{u}_2, \mathbf{W}_b, \mathbf{w}_b, \{cmt^k_b\}^t_{k=1}) \oplus F(\mathbf{C}^\top_b). \quad (7)$$

(g) $B$ also sets each element of $\overline{cmt}_b = (\overline{cmt}^1_b, \ldots, \overline{cmt}^t_a)$ as

$$\overline{cmt}^k_b = \begin{cases} \langle cmt^k_b(1) \rangle, & ch^k_b = 1; \\ \langle cmt^k_b(2) \rangle, & ch^k_b = 2; \\ \langle cmt^k_b(3) \rangle, & ch^k_b = 3. \end{cases} \quad (8)$$

(h) $B$ samples another noise $\tilde{\mathbf{E}}_b \hookleftarrow \chi(\mathbb{Z}^{m_1 \times m_1}_{q_1})$ and computes an auxiliary matrix $\mathbf{V}_b = \mathbf{S}_b \cdot \mathbf{C}'_a + \tilde{\mathbf{E}}_b$. Then $B$ generates the reconciliation matrix $\mathbf{M} \in \mathbb{Z}^{m_1 \times m_1}_2$ holds that

$$\mathbf{M}[i,j] = \lfloor \frac{2^{\theta+1}}{q_1} \cdot \mathbf{V}_b[i,j] \rfloor \mod 2, \quad \forall\, i,j \in [m_1], \quad (9)$$

where each entry of $V_b$ is viewed as an integer in $[-q_1/2, q_1/2-1]$[2].

(i) $B$ generates the shared session key $\mathbf{K}_b \in \mathbb{Z}^{m_1 \times m_1}_{2^\theta}$ by rounding the $\theta$ most significant bits from each entry of $\mathbf{V}_b$, i.e., the $(i,j)$-th entry of $\mathbf{K}_b$ is:

$$\mathbf{K}_b[i,j] = \lfloor \frac{2^\theta}{q_1} \cdot \mathbf{V}_b[i,j] \rceil \mod 2^\theta, \quad (10)$$

where entries of $\mathbf{V}_b$ are also viewed as integers in $[-q_1/2, q_1/2-1]$.

(j) $B$ sets $\mathtt{PROOF}_b = (\overline{cmt}_b, ch_b, \{rsp^k_b\}^t_{k=1}, \rho_b, \mathbf{w}_b, \mathbf{M})$ and authentication code $\mathtt{V}_b = \mathcal{H}_2(\mathbf{K}_b \| \mathbf{C}_b \| \mathbf{0})$. Then he sends $(\mathtt{PROOF}_b, \mathtt{V}_b)$ to $A$.

*Remark 1.* There is one pivotal modification of the above interactive algorithm: the transported tuple $\mathtt{PROOF}_a$ is a partial $\mathsf{NIZKAoK}$ compared with the original one generated in [18]. Namely, $B$ can recover 2/3 part $(cmt'_a)$ of the whole commitments $cmt_a$ from $ch_a$ and $rsp_a$, yet the validity of $cmt'_a$ cannot be verified since he only received the rest 1/3 contents $(\overline{cmt}_a)$. In this way, the only information $B$ can get is the retrieved message $\mathbf{C}'_a$. Therefore, $B$ is unable to detect which group $A$ belongs to in this flow, and has to symmetrically send his proof (masking his message $\mathbf{C}_b$) of group credential to $A$. This strategy also fix the flaw of Zhang *et al.*'s scheme [32].

[2] This can be done by setting $V_b[i,j]' = V_b[i,j] - \alpha q_1$ where $\alpha = 1$ if $V_b[i,j] > \frac{q_1}{2} - 1$ and $\alpha = 0$ otherwise.

3. $A \rightarrow B : (\mathtt{V}_a)$
   (a) $A$ sets $\mathbf{W}'_b = \mathcal{H}_1(\mathbf{A}, \mathbf{u}_1, \mathbf{K}, \rho_b)$. Then for each $\mathbf{v}_j \in \mathcal{CRL}_1$, $A$ computes $\mathbf{e}'_j = \mathbf{w}_b - \mathbf{W}'_b \cdot \mathbf{v}_j$. If there exist an index $j$ such that $\|\mathbf{e}'_j\|_\infty \leq \beta$, $A$ chooses a random value $\mathtt{V}_a \xleftarrow{\$} \{0,1\}^{q_1}$, outputs 0 and aborts.
   (b) $A$ also computes the checked value $cmt^*_b = (cmt^{*1}_b, \ldots, cmt^{*t}_b)$ for $k \in [t]$ from corresponding $rsp^k_a$ and $ch^k_a$ as follows:

$$cmt^{*k}_b = \begin{cases} \langle cmt^{*k}_b(2), cmt^{*k}_b(3) \rangle, & ch^k_b = 1; \\ \langle cmt^{*k}_b(1), cmt^{*k}_b(3) \rangle, & ch^k_b = 2; \\ \langle cmt^{*k}_b(1), cmt^{*k}_b(2) \rangle, & ch^k_b = 3. \end{cases} \tag{11}$$

   Then with identical operations $A$ recovers the original $cmt'_b$ and computes the masked matrix $\mathbf{C'}_b^{\top} = F^{-1}(ch_b \oplus \mathcal{H}_0(\mathbf{A}, \mathbf{u}_1, \mathbf{W}'_b, \mathbf{w}_b, cmt'_b))$.
   (c) $A$ computes an assistant matrix $\mathbf{V}_a = \mathbf{C}'_b \cdot \mathbf{S}_a$. Next, she extracts the shared session key $\mathbf{K}_a \in \mathbb{Z}_{2^\theta}^{m_1 \times m_1}$ from $\mathbf{V}_a$ via a reconciliation technique, i.e., using the check field $\mathbf{M}$ to apply the rounding. The $(i,j)$-th entry of $\mathbf{K}_a$ is:

$$\mathbf{K}_a[i,j] = \lfloor \frac{2^\theta}{q_1} \cdot \mathbf{V}_a[i,j] + \frac{1}{4} \cdot (2\mathbf{M}[i,j] - 1) \rceil \mod 2^\theta, \tag{12}$$

   where each entry of $V_b$ is also viewed as an integer in $[-q_1/2, q_1/2 - 1]$.
   (d) $A$ verifies that $\mathtt{V}_b \stackrel{?}{=} \mathcal{H}(\mathbf{K}_a \| \mathbf{C}'_b \| \mathbf{0})$. If so, $A$ outputs 1 and sends $\mathtt{V}_a = \mathcal{H}_2(\mathbf{K}_a \| \mathbf{C}_a \| \mathbf{1})$ to $B$. Otherwise, $A$ outputs 0 and responds a random $\mathtt{V}_a$.
   (e) $B$ verifies $\mathtt{V}_a$ through a similar equation $\mathtt{V}_a \stackrel{?}{=} \mathcal{H}(\mathbf{K}_b \| \mathbf{C}'_a \| \mathbf{1})$. $B$ outputs 1 if the equation holds, else he outputs 0.

- TraceMember: When a dispute happens, firstly GA will retrieve the handshake transcripts of $A$ and $B$. Then for $d \in [N]$, GA computes $\mathbf{e}_d = \mathbf{w} - \mathbf{W} \cdot \mathsf{urt}[d]$ and outputs the first index $d^*$ such that $\|\mathbf{e}_{d^*}\|_\infty \leq \beta$, otherwise outputs $\perp$ indicating that the involved participant is a malicious outsider.
- RemoveMember: GA maintains and updates the information of $\mathcal{CRL}$ and $\mathcal{L}$ after tracing a malicious group member or receiving a logout request. To remove a member $U$ from group $G$, GA first looks up and removes the member's $\mathsf{UserSecret} = (U, \mathsf{ID}_U)$ from $\mathcal{L}$. Then GA adds $\mathsf{urt}[d_u]$ to $\mathcal{CRL}$, and distributes the updated list $\mathcal{CRL}$ to every other group members via an authenticated anonymous channel.

## 5 Security and Performance Analysis of the Scheme

### 5.1 Security

*Completeness*: We first demonstrate that the scheme is complete with overwhelming probability if both active users belong to the same group

($\mathsf{gpk}_1 = \mathsf{gpk}_2$), based on the perfect completeness of the underlying Stern-like protocol.

Note that the interactive message PROOF generated by an honest and active user is always valid, i.e., the receiver can rightly recover the original commitments holding that $cmt' = cmt$. Thus, such a receiver can always retrieve the hidden matrix $\mathbf{C}'$ equals to $\mathbf{C}$. In this way, it can be deduced from [8, A.1.4] that $\mathbf{K}_a \neq \mathbf{K}_b$ with negligible probability. Therefore, both authentication codes ($\mathtt{V}_a$ and $\mathtt{V}_b$) would be successfully verified, and consequently the handshake protocol outputs 1 for $A$ and $B$. Moreover, the tracing algorithm $\mathsf{TraceMember}$ will get $\mathbf{e}_{d^*} = \mathbf{w} - \mathbf{W} \cdot \mathsf{urt}[d^*] = \mathbf{e}^*$ for $d^* = d_a$ or $d_b$, where $\mathbf{e}^*$ is sampled from a $\beta$-bounded distribution such that $\|\mathbf{e}^*\|_\infty \leq \beta$ holds with overwhelming probability. So both users can always be traced when disputes happen.

*Privacy*: We now prove that our scheme satisfies the privacy requirements listed in Sect. 3 through Theorems 1–3 below, for which some proofs are deferred to Appendix 1.

**Theorem 1.** *In the random oracle model, impersonator resistance holds for our scheme under the* $\mathsf{SIS}$ *assumption.*

**Theorem 2.** *In the random oracle model, detector resistance holds for our scheme under the* $\mathsf{LWE}$ *assumption.*

**Theorem 3.** *In the random oracle model, unlinkability holds for our scheme under the* $\mathsf{LWE}$ *assumption.*

*Proof.* The proof is similar to that of Theorem 2. Based on the security of utilized $\mathsf{ZK}$ protocol and the $\mathsf{LWE}_{n,q,\chi}$ assumption, we can build a sequence of games to argue that $|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{Unlink}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{Unlink}-0} = 1]| = \mathsf{negl}(\lambda)$.

### 5.2 Performance

From Theorem 1, 2 and 3 we know that our scheme's provable security depends on the $\mathsf{LWE}_{n,q,\chi}$ and $\mathsf{SIS}^{\infty}_{n,(\ell+1)\cdot m,q,2\beta}$ (implying $\mathsf{SIS}^{2}_{n,(\ell+1)\cdot m,q,2\beta\sqrt{(\ell+1)\cdot m}}$) assumptions. Recently Yang *et al.* [31] gave a concrete technique to estimate and derive parameters for hardness theorems over lattices. They examine the root Hermite factor (RHF) and summarize the required RHF for these problems as follows:

$$\mathrm{RHF} = \begin{cases} \exp(\frac{\log^2 \beta}{4n \log q}), & \text{for } \mathsf{SIS}^{2}_{n,m,q,\beta}; \\ \exp(\frac{\log^2 \frac{\sigma \cdot \sqrt{2\pi}}{5.31q}}{4n \log q}), & \text{for } \mathsf{LWE}_{n,q,\chi}. \end{cases} \tag{13}$$

We adopt this method and set RHF as 1.0048 to achieve an 80-bit security. In this way, we get $n = 471$ from the above equations. Then we set

$$(q, m, \sigma, \beta, m_1, q_1, \theta, \sigma_1) = (1961767, 19654, 296, 389, 6, 2^{13}, 4, 2309),$$

according to the asymptotic bounds of these scheme parameters. To make soundness error of the $\mathsf{ZK}$ protocol negligible, we set repetitions $t = 137$. Besides, we set $N = 2^{19}$ for efficient group management.

- **Communication Cost:** In Addmember, GA sends $\mathsf{Cred}_u$ to the user, where $\mathsf{ID}_u = d_u$ is a binary string of size $\ell$, $(\mathsf{usk}[d_u], \mathsf{urt}[d_u])$ comprises an element of $\mathbb{Z}^{(2\ell+1)m}$ holding $\|\mathsf{usk}[d_u]\|_\infty \leq \beta$ and an element of $\mathbb{Z}_q^n$, respectively. So the communication cost in this step is less than $\ell + (\ell+1)m \log \beta + n \log q$ bits ≈579 KB. In Handshake, each member finally needs to transmit a pair $(\mathtt{PROOF}, \mathtt{V})$ in two rounds. PROOF comprises partial commitments $\overline{cmt}$ with bit-size $tn \log q$, a challenge value $ch$ having $2t$ bits, $t$ responses $rsp$ with no more than $3tm \log \beta(4\ell + 4 + 2(\ell+1) \log q)$ bits and an LWE function output $\mathbf{w} \in \mathbb{Z}_q^m$, so the total length is about 7.79 GB (user $B$ sends an extra matrix $\mathbf{M} \in \mathbb{Z}_2^{m_1 \cdot m_1}$ having bit-size 36). The authentication code V has length $n \log q$ bits. Thus, the communication cost in Handshake for each participant is about 7.8 GB.
- **Computational Cost:** In CreateGroup, GA generates all the potential users' credentials, the main computation cost here is the polynomial-time algorithms GenTrap and SamplePre which can be pre-computed. In Handshake, the main operations here are the multiplications of matrices and vectors in $\mathcal{O}(n^2)$ and a polynomial-time commitment scheme COM [16] used in ZK protocol, whose runtime is on the order of milliseconds using libraries like GMP [11] and NTL [24]. Thus the computational cost of Handshake is considered to be very small. While the step of revocation check runs in the size of list $\mathcal{CRL}$, as it seems unavoidable for the setting of VLR.

## 6 Conclusion

This paper aims to propose the first secret handshake scheme from lattices, which supports reusable credentials and membership revocation. With some subtle modifications, we transform a Stern-like ZKAoK system into a mutual authentication protocol. It's intriguing to consider whether this design is a generic framework, e.g., can apply to other types of ZKAoK like Fiat-Shamir with abort ones [21,31]. To achieve traceability and unlinkability with ease, we utilize the VLR structure to generate revocation tokens with group identities encoded in a Bonsai tree. We believe that, our construction - while not being entirely novel - would certainly help to exploit the area of post-quantum secret handshakes. One interesting future work is to build secret handshakes supporting more functionalities such as backward unlinkability or full dynamicity through more efficient lattice-based techniques.

**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).

## Appendix 1. Impersonator Resistance (Proof of Theorem 1)

*Proof.* Suppose that $\mathcal{A}$ succeeds in experiment $\mathbf{Exp}^{\mathsf{IR}}_{\mathcal{A}}$ with non-negligible advantage $\epsilon$. Then we can build a PPT algorithm $\mathcal{F}$ that solves $\mathsf{SIS}^{\infty}_{n,(l+1)\cdot m,q,2\beta}$ problem with non-negligible probability.

Given an $\mathsf{SIS}$ instance $\mathbf{C} = [\mathbf{C}_0|\mathbf{C}_1|\dots|\mathbf{C}_\ell] \in \mathbb{Z}_q^{n\times(\ell+1)m}$, the goal of $\mathcal{F}$ is to find a non-zero vector $\mathbf{y} \in \mathbb{Z}^{(\ell+1)\cdot m}$ such that $\mathbf{C}\cdot\mathbf{y} = \mathbf{0} \mod q$ and $\|\mathbf{y}\|_\infty \le 2\beta$. Toward this goal, $\mathcal{F}$ first generates the public parameters $\mathsf{par}$ as we do in $\mathsf{Setup}$, and proceeds as described in experiment $\mathbf{Exp}^{\mathsf{IR}}_{\mathcal{A}}$. Note that $\mathcal{F}$ can consistently answer all the oracle queries made by $\mathcal{A}$. In particular, $\mathcal{F}$ randomly picks $i \in [q_G]$ where $q_G$ is the number of queries to oracle $\mathsf{KeyP}$, then it performs the following steps at the $i$-th query to oracle $\mathsf{KeyP}$ to bulid a group $G^{(i)}$:

- Sample vector $\mathbf{z} = (\mathbf{x}_0|\mathbf{x}_1|\dots|\mathbf{x}_\ell) \in \mathbb{Z}^{(\ell+1)\cdot m}$ from $D_{\mathbb{Z}^{(\ell+1)\cdot m},\sigma}$. If $\|\mathbf{z}\|_\infty > \beta$, repeat the sampling. Otherwise, compute $\mathbf{u} = \mathbf{C}\cdot\mathbf{z} \mod q$.
- Get $\ell$ pairs $\{(\mathbf{F}_i, \mathbf{R}_i)\}_{i\in[\ell]}$ by invoking algorithm $\mathsf{GenTrap}(n, m, q)$ for $\ell$ times.
- Choose a target identity $d^* \xleftarrow{\$} \{0,1\}^\ell$, and define $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\dots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n\times(2\ell+1)m}$ by setting $\mathbf{A}_0 = \mathbf{C}_0$, $\mathbf{A}_i^{d^*[i]} = \mathbf{C}_i$ and $\mathbf{A}_i^{1-d^*[i]} = \mathbf{F}_i$ for $i \in [\ell]$.
- Define the secret key and revocation token of member $d^*$ as follows:
  - i: $\mathsf{usk}[d^*] = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\dots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, where $\mathbf{x}_0 = \mathbf{z}_0$, $\mathbf{x}_i^{d^*[i]} = \mathbf{z}_i$ and $\mathbf{x}_i^{1-d^*[i]} = \mathbf{0}^m$ for all $i \in [\ell]$.
  - ii: $\mathsf{urt}[d^*] = \mathbf{A}_0\cdot\mathbf{x}_0 \mod q \in \mathbb{Z}_q^n$.
- For member's identity $d \ne d^*$, generate its secret key and revocation token as follows:
  1. Since $d \ne d^*$, there exists an index $p$ being the first index of *LTR*-order such that $d[p] \ne d^*[p]$. Then it holds that $\mathbf{A}_p^{d[p]} = \mathbf{A}_p^{1-d*[p]} = \mathbf{F}_p$.
  2. Sample $\ell$ vectors $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_{p-1}^{d[p-1]}, \mathbf{x}_{p+1}^{d[p+1]}, \dots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, and set $\mathbf{s}^{(d)} = \mathbf{u} - (\mathbf{A}_0\cdot\mathbf{x}_0 + \sum_{i\in[\ell],i\ne b}(\mathbf{A}_i^{d[i]}\cdot\mathbf{x}_i^{d[i]})) \mod q$.
  3. Sample $\mathbf{x}_p^{d[p]} \hookleftarrow \mathsf{SamplePre}(\mathbf{R}_p, \mathbf{F}_p, \mathbf{s}^{(d)}, \sigma)$.
  4. Set $\mathbf{x}^{(d)} = (\mathbf{x}_0\|\mathbf{x}_1^0\|\mathbf{x}_1^1\|\dots\|\mathbf{x}_\ell^0\|\mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$, where $\mathbf{x}_i^{1-d[i]} = \mathbf{0}^m$ for all $i \in [\ell]$. Repeat the sampling if $\|\mathbf{x}^{(d)}\|_\infty > \beta$. Otherwise, let $\mathsf{usk}[d] = \mathbf{x}^{(d)}$ and $\mathsf{urt}[d] = \mathbf{A}_0\cdot\mathbf{x}_0 \mod q$.
- Set $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, $\mathsf{gsk} = (\mathbf{R}_i, \mathsf{grt})$, and $\mathsf{usk} = \{\mathsf{usk}[k]\}_{k=1}^N$. Note that, by construction, the distribution of $(\mathsf{gpk}, \mathsf{grt}, \mathsf{usk})$ is statistically close to that of the real scheme, and the choice of $d^*$ is hidden from the adversary.

Eventually, $\mathcal{A}$ wins with its output $\mathtt{PROOF}^* = (\overline{cmt}^*, ch^*, \{rsp_k^*\}_{k=1}^t, \rho^*, \mathbf{w}^*)$. Since the involved user outputs 1 after a handshake with $\mathcal{A}$, we know that he must have retrieved the right hidden matrix $\mathbf{C}^*$. This fact also means that the recovered commitments $cmt'^*$ is equal to the original one $cmt^*$. Now it can be deduced that the $\mathsf{NIZKAoK}$ $(cmt^*, ch^*, \{rsp_k^*\}_{k=1}^t)$ is a valid one generated by $\mathcal{A}$ via the underlying $\mathsf{ZK}$ protocol. Then we can argue that $\mathcal{A}$ must have queried

$\mathcal{H}_0$ on input $(\mathbf{A}, \mathbf{u}, \mathbf{W}^*, \mathbf{w}^*, cmt^*)$ (denoted as $\eta^*$), as otherwise, the probability that $ch^* = \mathcal{H}_0(\eta^*)$ is at most $3^{-t}$. Thus, with probability at least $\epsilon - 3^{-t}$, there exists some $\kappa^* \leq q_{\mathcal{H}}$ such that the $\kappa^*$-th hash query involves the tuple $\eta^*$, where $q_{\mathcal{H}}$ is the number of queries to random oracle $\mathcal{H}_0$.

To employ the Improved Forking Lemma [22], $\mathcal{F}$ reinvokes $\mathcal{A}$ polynomial times with the same random tape and input as in the original run, until the $\kappa^*$ query, that is, from the $\kappa^*$ query onwards, $\mathcal{F}$ answers $\mathcal{A}$ with fresh and independent values $\rho_{\kappa^*}, \ldots, \rho_{q_{\mathcal{H}}} \xleftarrow{\$} \{1,2,3\}^t$. By the aforementioned Forking Lemma, with probability $\geq \frac{1}{2}$, $\mathcal{F}$ obtains 3-fork $\{\rho^1_{\kappa^*}, \rho^2_{\kappa^*}, \rho^3_{\kappa^*}\}$ involving the same tuple $\eta^*$ after less than $32 \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t})$ executions of $\mathcal{A}$. Then we have $\{\rho^1_{\kappa^*}(i), \rho^2_{\kappa^*}(i), \rho^3_{\kappa^*}(i)\} = \{1,2,3\}$ for some $i \in [t]$ with probability $1 - (\frac{7}{9})^t$. Having such index $i$, $\mathcal{F}$ can parse the 3 forgeries from the fork branches to obtain 3 valid responses $(rsp^*_i(1), rsp^*_i(2), rsp^*_i(3))$ w.r.t. 3 different challenges for the same commitment $cmt^*_i$. By Theorem 1 in [18], we can extract vectors $\mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}^0_1 \| \mathbf{x}^1_1 \| \ldots \| \mathbf{x}^0_\ell \| \mathbf{x}^1_\ell) \in \mathbb{Z}^{(2\ell+1)m}$ and $\mathbf{e}^* \in \mathbb{Z}^m$ such that:

1. $\|\mathbf{x}\|_\infty \leq \beta$, the following $\ell$ blocks are zero-blocks $\mathbf{0}^m$: $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ for some $d \in \{0,1\}^\ell$;
2. $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$;
3. $\|\mathbf{e}'\|_\infty \leq \beta$ and $\mathbf{w}^* = \mathbf{W}^* \cdot (\mathbf{A}_0 \cdot \mathbf{x}_0) + \mathbf{e}^* \mod q$.

Now we consider two cases:

- If $G^*$ is not created at the $i$-th query to oracle $\mathsf{KeyP}$ or $d \neq d^*$, which happens with probability at most $\frac{N \cdot q_G - 1}{N \cdot q_G}$, then algorithm $\mathcal{F}$ fails and aborts.
- If $d = d^*$ belongs to $G^{(i)}$, set $\mathbf{x}^* = (\mathbf{x}_0 \| \mathbf{x}_1^{d[1]} \| \ldots \| \mathbf{x}_\ell^{d[\ell]}) \in \mathbb{Z}^{(\ell+1)m}$. Then by construction it holds that $\mathbf{C} \cdot \mathbf{x}^* = \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \mod q$. Furthermore, experiment $\mathbf{Exp}^{\mathsf{IR}}_{\mathcal{A}}$ ensures that $\mathcal{A}$ has never requested the user secret key $\mathsf{usk}[d^*]$, so that $\mathbf{z}$ is unknown to $\mathcal{A}$. In this case, because $\mathbf{z}$ has large min-entropy given $\mathbf{u}$ (see Lemma 1), we have $\mathbf{x}^* \neq \mathbf{z}$ with overwhelming probability.

  Now let $\mathbf{y} = \mathbf{x}^* - \mathbf{z}$, then we get the following facts: i) $\mathbf{y} \neq \mathbf{0}$; ii) $\mathbf{C} \cdot \mathbf{y} = \mathbf{0} \mod q$; iii) $\|\mathbf{y}\|_\infty \leq \|\mathbf{x}^*\|_\infty + \|\mathbf{z}\|_\infty \leq \beta + \beta = 2\beta$. So $\mathcal{F}$ finally outputs the vector $\mathbf{y}$, which is a solution to the related $\mathsf{SIS}^\infty_{n,(l+1)\cdot m,q,2\beta}$ problem.

In summary, the probability that $\mathcal{F}$ does not abort and solve the $\mathsf{SIS}^\infty_{n,(l+1)\cdot m,q,2\beta}$ assumption is larger than $(1 - (\frac{7}{9})^t)/2(N \cdot q_G)$. This concludes the proof.

## Appendix 2. Detector Resistance (Proof of Theorem 2)

*Proof.* We define a sequence of hybrid games where the first is $\mathbf{Exp}^{\mathsf{DR}-0}_{\mathcal{A}}$ and the last is $\mathbf{Exp}^{\mathsf{DR}-1}_{\mathcal{A}}$. Then we prove that these games are indistinguishable. For $i$-th game, denote the output of $\mathcal{A}$ by $R_i$. The concrete games are described as follows.

**Game** 0**:** This is exactly the original game $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-0}$.
**Game** 1**:** This game is the same as Game 0 except that it generates a simulated proof for the interactive handshake between $\mathcal{A}$ and the chosen user $\mathsf{ID}^*$, via running the simulator of the underlying argument for every repetition, and then generates the corresponding challenge via oracle $\mathcal{H}_0$. Since the hidden vector $\mathbf{C}$ is also generated randomly by an $\mathsf{LWE}$ function, the view of adversary $\mathcal{A}$ is statistically indistinguishable between Game 1 and Game 2 by zero-knowledge property of underlying $\mathsf{ZK}$ protocol. So $\Pr[R_1 = 1] \approx \Pr[R_2 = 1]$.
**Game** 2**:** This game is the same as Game 1 with only one modification: for token embedding, we compute the $\mathsf{LWE}$ function using a random nonce $\mathbf{s}$ instead of the revocation token $\mathsf{urt}[\mathsf{ID}^*]$, namely, $\mathbf{w} = \mathbf{W} \cdot \mathbf{s} + \mathbf{e}^* \mod q$ where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. Recall that the token $\mathsf{urt}[\mathsf{ID}^*] = \mathbf{A}_0 \cdot \mathbf{x}_0$ is statistically close to uniform over $\mathbb{Z}_q^n$. In this way, we have $\Pr[R_2 = 1] \approx \Pr[R_1 = 1]$.
**Game** 3**:** This game follows Game 2 with one change: we make $\mathbf{w}$ uniformly sampled 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, it holds that $\Pr[R_3 = 1] - Pr[R_2 = 1] = \mathsf{negl}(\lambda)$.
**Game** 4**:** This game switches back to use a random nonce to produce $\mathbf{w}$, and this $\mathsf{LWE}$ function is for an arbitrary user $\mathsf{ID}_r$, i.e., $\mathbf{w} = \mathbf{W} \cdot \mathbf{s} + \mathbf{e}_r \mod q$. Since $\mathbf{e}_r \hookleftarrow \chi^m$ is $\beta$-bounded, the output $\mathtt{PROOF}$ is computationally close to that in Game 3. Hence we have $\Pr[R_4 = 1] - Pr[R_3 = 1] = \mathsf{negl}(\lambda)$.
**Game** 5**:** In this game, we generate $\mathbf{w}$ with another user's revocation token $\mathsf{urt}[\mathsf{ID}_r]$, namely, $\mathbf{w} = \mathbf{W} \cdot \mathsf{urt}[\mathsf{ID}_r] + \mathbf{e}_r \mod q$. Since $\mathsf{urt}[\mathsf{ID}_r]$ is statistically close to uniform over $\mathbb{Z}_q^n$, this change makes no difference to the view of $\mathcal{A}$. Therefore, it holds that $\Pr[R_5 = 1] \approx \Pr[R_4 = 1]$.
**Game** 6**:** This game is exactly the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{DR}-1}$. We generate the real argument for the handshake between $\mathcal{A}$ and $\mathsf{ID}_r$, the transcript is statistically indistinguishable from that of Game 5 by the zero-knowledge property of the utilized $\mathsf{ZKAoK}$. In this way, we have $\Pr[R_6 = 1] \approx \Pr[R_5 = 1]$.

Combining the above analysis, we have 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.

## References

1. 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
2. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_1
3. 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
4. Ateniese, G., Kirsch, J., Blanton, M.: Secret handshakes with dynamic and fuzzy matching. In: NDSS 2007. The Internet Society (2007)

5. 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
6. 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
7. 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
8. ETSI: ETSI TR 103 570: CYBER; Quantum-Safe Key Exchange, 1.1.1 edn. (2017)
9. 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
10. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: Sedgewick, R. (ed.) STOC 1985, pp. 291–304. ACM (1985). https://doi.org/10.1145/22145.22178
11. Granlund, T.: The GMP Development Team: GNU MP: The GNU Multiple Precision Arithmetic Library, 6.1.2 edn. (2016). http://gmplib.org/
12. He, D., Kumar, N., Wang, H., Wang, L., Choo, K.R., Vinel, A.V.: A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network. IEEE Trans. Dependable Secur. Comput. **15**(4), 633–645 (2018). https://doi.org/10.1109/TDSC.2016.2596286
13. 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
14. 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
15. 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
16. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_23
17. Kulshrestha, P., Pal, A.: A new secret handshakes scheme with dynamic matching based on ZSS. IJNSA **7**(1), 67–78 (2015)
18. Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 345–361. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_20
19. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 1–31. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_1
20. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41

21. del Pino, R., Lyubashevsky, V., Seiler, G.: Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) CCS 2018, pp. 574–591. ACM (2018). https://doi.org/10.1145/3243734.3243852
22. Pointcheval, D., Vaudenay, S.: On provable security for digital signature algorithms. Technical report LIENS-96-17 of the Laboratoire d'Informatique de Ecole Normale Superieure, November 1996
23. 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
24. Shoup, V.: A Tour of NTL, 11.4.3 edn. http://www.shoup.net/ntl/
25. 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
26. Tsudik, G., Xu, S.: A flexible framework for secret handshakes. In: Danezis, G., Golle, P. (eds.) PET 2006. LNCS, vol. 4258, pp. 295–315. Springer, Heidelberg (2006). https://doi.org/10.1007/11957454_17
27. 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
28. Wen, Y., Zhang, F.: Delegatable secret handshake scheme. J. Syst. Softw. **84**(12), 2284–2292 (2011). https://doi.org/10.1016/j.jss.2011.06.046
29. Wen, Y., Zhang, F., Xu, L.: Secret handshakes from id-based message recovery signatures: a new generic approach. Comput. Electr. Eng. **38**(1), 96–104 (2012). https://doi.org/10.1016/j.compeleceng.2011.11.020
30. 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
31. 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
32. 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
33. 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