Group Signature with Verifier-Local Revocation Based on Coding Theory

Luping Wang,1 Kai Zhang,2 Haifeng Qian,1,3 and Jie Chen1,4

1East China Normal University, Shanghai, China
2Shanghai University of Electric Power, Shanghai, China
3Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China
4Shanghai Key Laboratory of Trustworthy Computing, Shanghai, China

Correspondence should be addressed to Jie Chen; s080001@e.ntu.edu.sg

Received 13 April 2021; Revised 6 July 2021; Accepted 17 July 2021; Published 21 August 2021

Abstract

Group signature with verifier-local revocation (VLR-GS) is a special variant of revocable group signature that not only allows a user to anonymously sign messages but also only requires the verifiers to possess some up-to-date revocation information. To date, a number of VLR-GS schemes have been proposed under bilinear groups and lattices, while they have not yet been instantiated based on coding theory. In this paper, we present a code-based VLR-GS scheme in the random oracle model, which is the first construction to the best of our knowledge. Concretely, our VLR-GS scheme does not rely on the traditional paradigm which utilizes an encryption scheme as a building block and achieves logarithmic-size group signature. To obtain the scheme, we first introduce a new code-based Stern-like interactive zero-knowledge protocol with member revocation mechanism based on syndrome decoding problem. Moreover, we employ the binary Goppa code embedded for our scheme with efficiency and security analysis.

1. Introduction

1.1. Group Signature. Group signature scheme is a fundamental cryptographic primitive that enables signer anonymity and has been extensively researched since introduced by Chaum and Van Heyst [1]. The anonymity indicates that any group member is allowed to issue signatures for documents on behalf of the group, while the identity of the signer is anonymously hidden from a verifier. Concretely, it allows the group members to issue signatures on behalf of the whole group without compromising their identity information. When receiving an issued signing key from the group manager, a signer can generate a group signature on a certain message. Moreover, a verifier can verify the signature by only a group public key and cannot identify who is the actual signer even when the validity of the signature is verified.

At the same time, there is a serious problem that should be considered in the real-world applications. If a group member issues signatures for documents where he is not allowed to sign, this member should be revoked from the group. The membership of a revoked user should be canceled. That is to say, how to remove the signing capability of misbehaving users without affecting remaining members is a significant problem. This problem can be encountered in many application scenarios, such as in politics and business. For example, if illegal information sent by a malicious group member is reported, the group administrator will kick the malicious member out of the group after verification to ensure the safety of other legitimate members in the group. A naive method is to rebuild the group signature system for distributing system parameter to the unrevoked members; however, this certainly limits the practicality. Hence, support of membership revocation in group signature is highly desirable, which makes the notion of revocable group signature (RGS) to be proposed [2] and widely researched.
1.2. Revocable Group Signature. When considering member revocation, how to efficiently and securely revoke the signing ability of misbehaving members in an unrevoked user-unaware way is a major problem in group signature.

Traditionally, the revocation information should be sent to both signer and verifier in the revocable group signature schemes [3–5]. Later, the most employed method is the verifier-local revocation technique [6], which only requires the verifier to possess some up-to-date revocation information, that is, the revocation messages are only sent to signature verifier, not the signer. Hence, the VLR-GS is more widely deployed in real-world application due to its high efficiency and flexible operations.

Currently, a number of VLG-GS schemes have been proposed based on various security assumptions in the pairing-free, pairing, and lattice setting [7–10], while they have not been studied yet based on coding theory. So, in this paper, we give a generic construction of the code-based VLG-GS scheme, which can resist quantum computer attacks.

1.3. Our Contributions. Our main contribution is to give a generic construction for the code-based group signature scheme with support of verifier-local revocation function. For this purpose, we first design a code-based Stern-like interactive zero-knowledge (ZK) identification protocol with member revocation mechanism; this protocol allows the prover to convince the given ciphertext is well formed, and the hidden plaintext satisfies other additional conditions. Then, we utilize the new ZK protocol to construct the code-based revocable group signature scheme. Our structure departs from the traditional “sign-and-encrypt-and-prove” group signature paradigm and deviates from code-based encryptions. So, how to construct an interactive ZK protocol with member revocation mechanism under a code-based security assumption is the main problem we have solved. Finally, we give a formal security proof for its correctness, selfless anonymity (this is a slightly weaker notion than the CPA anonymity and CCA anonymity for group signature, where this work focuses on providing membership revocation functionality for group signature), and traceability and reduce its security to the hardness of the syndrome decoding (SD) problem. By instantiating the scheme with $[n, k]_2$ Goppa codes with parameter analysis, the public key size and the signature size in the scheme achieve $\log(N) \cdot O(n^2)$ and $\log(N) \cdot O(n)$ over $\mathbb{F}_2$, respectively, where $N$ is the number of group users.

1.4. Technique Overview. The main building block of our VLR group signature scheme is an interactive ZK protocol. The protocol allows a prover to convince the verifier that he is a certified group member (i.e., he possesses a valid secret signing key) and that he has not been revoked (i.e., his “revocation token” is not in the verifier’s blacklist). We can repeat the protocol many times to make the soundness error negligibly small.

We consider a group of size $N = 2^l$, and each user is identified by a string $d = (d[1], \ldots, d[l]) \in \{0, 1\}^l$ denoting the binary representation of his index in the group. To get an extension of $d$, inspired by the Bonsai tree signature originally proposed by Cash et al. in [11], which was first introduced into the code theory by Ezerman et al. [12], here we adapt and develop their method to manage the Bonsai tree of a difficult code-based problem, and hence a group user with identity $d$ is issued a Bonsai signature under its identity. That is, we have a small vector $z \in \mathbb{F}_2^{(2l^2+n)}$ with $z$’s Hamming weight $w(z) = r$ and $A \cdot z = u$, where $A_d = [\hat{A}_1, \ldots, \hat{A}_d] \ldots [\hat{A}_d]$ is a subtree defined by $d$.

The anonymity implies that the signer’s identity $d$ is hidden from the verifier; therefore, the matrix $A_d$ should not be explicitly given. We utilize the extension method in [7]: we add $n$ suitable zero blocks to vector $z$ to obtain an extended vector $x = (x_0, x_1, \ldots, x_l) \in \mathbb{F}_2^{(2l^2+n)}$, where the added zero blocks are $x_1, \ldots, x_l$. We then have $w(x) = r$, and $A \cdot x = u$. Namely $x$ is a solution to the syndrome decoding instance given by the whole Bonsai tree. To prove in zero-knowledge the possession of such a vector $x$, we adapt the “Stern extension” proof system, and the user identity $d$ is hidden by a “one-time pad” technique. As a result, the user can anonymously prove his group membership.

To enable membership revocation, the secret key $x$ of each group member should make the first block $x_0$ related with $A_0$; then, the revocation token could be formulated as $A_0 \cdot x_0 \in \mathbb{F}_2^2$. With well-chosen parameters and sample $x_0$ from a proper distribution, the token can therefore be statistically close to uniform over $\mathbb{F}_2^n$. In fact, when a user is asked to sample a uniformly random vector $r_0 \in \mathbb{F}_2^n$ and to compute a commitment $c_0$ using a (code-based) collision-resistant hash function COM, for which the value $A_0 \cdot r_0$ is part of the committed string. Depending on the verifier’s challenge, the user will either reveal $r_0$ or $r_0 + x_0$.

1.5. Related Work. Group signature schemes have been well studied based on bilinear pairing [6, 13–17]. In 2018, Ishida et al. and Perera and Koshiba [18, 19] gave two group signature frameworks, respectively, which accorded to the traditional “sign-and-encrypt” paradigm and could achieve a high level of anonymity: (almost) full anonymity. However, because the secure code-based signature scheme cannot be realized for a while, the code-based instantiation scheme cannot be presented. Later, a number of lattice-based and code-based group signature schemes have been proposed, which can resist quantum computer attacks. Gordon et al. [20] designed the first lattice-based group signature scheme whose public key size and signature size are linear in the number of group members $N$. Recently, there are several efficient lattice-based constructions [7, 21, 22] that have been proposed, whose group public key size and signature size are proportional to $\log(N)$. The first code-based group signature scheme was proposed by Alamolh et al. in [23], which allows to dynamically add new members during the lifetime of the group. In 2015, Ezerman et al. [12] proposed the first provably secure group signature scheme from code-based assumptions which satisfies CPA anonymity and traceability requirements in the random oracle model, whose security is reduced to the hardness of the syndrome decoding problem.
Moreover, the first rank code-based group signature was proposed by Alamélou et al. [24], which achieves dynamic in a relaxed pattern based on the BSZ model of Bellare et al. [13]. Nevertheless, there is no instantiation code-based group signature scheme with support of revocation functionality.

The concept of VLR-GS was first considered by Brickell [25] and Kiayias et al. [26] and then formalized by Boneh and Shacham [6]. The state-of-the-art VLR-GS constructions are usually given in the bilinear map. The first lattice-based VLR-GS scheme was given by Langlois et al. [7]. In 2020, Yin et al. [27] proposed a lattice-based VLR-GS scheme and realized full anonymity. However, this property cannot be implemented in code-based group signature temporarily, for the research about code cryptography is not mature as lattice’s.

1.6. Organization. Section 2 recalls some preliminary knowledge for our work, including code-based cryptography and group signature. In Section 3, we introduce a code-based “Stern-like” interactive protocol, which supports member revocation. We construct a code-based VLR group signature scheme and give its security analysis in Section 4. In the end, we conclude our work in Section 5.

2. Preliminaries

In this section, we present some notations and technical tools used in this work.

2.1. Notations. If $\mathcal{D}$ is a finite set, $s^\mathcal{D}$ denotes the operation of picking an element of $\mathcal{D}$ uniformly at random and assigning the result to a variable $s$. Let $N$ be a positive integer, and we denote $[N] \rightarrow \{1, \ldots, N\}$. Let $B(n, r)$ be the set of all vectors $x \in \mathbb{F}_2^n$ such that $w(x) = r$, where $w(x)$ is the Hamming weight of vector $x$. In addition, we use bold lowercase and capital letters to denote vectors and matrices, respectively.

2.2. General Definitions

**Definition 1** (linear codes). A $[n, k] \ (n > k)$ linear error-correcting code $C$ over a finite field $\mathbb{F}_2$ is a $k$-dimensional subspace of $n$-dimensional linear space $\mathbb{F}_2^n$, and co-dimension is $n-k$. Let matrix $G \in \mathbb{F}_2^{(n-k)\times n}$ be a generator matrix of code $C$, and the row vectors of $G$ form a base of $C$, i.e., $C = \{m \cdot G \in \mathbb{F}_2^n | m \in \mathbb{F}_2^{(n-k)}\}$.

Note that $G$ is not unique. In addition, the code $C$ can also be defined by a parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ such that $C = \{c \in \mathbb{F}_2^n | H \cdot c^\top = 0^{n-k}\}$, and hence the check matrix of code $C$ is not unique as well. Besides, any row vector of $H$ is orthogonal to any row vector of the generating matrix $G$ of $C$, i.e., $H \cdot G^\top = 0$. For any vector $x \in \mathbb{F}_2^n$, define the syndrome of $x$ as $s = H \cdot x^\top \in \mathbb{F}_2^{n-k}$, where $x$ is a codeword of $C$ if and only if $H \cdot x^\top = 0^{n-k}$.

Besides, we say that $G$ (respectively, $H$) is in systematic form if it is of the form $(I_k | A)$ (respectively, $(I_{n-k} | B)$).

**Definition 2** (syndrome decoding problem, SDP). Let $H$ be a parity check matrix for a random $[n, k]$ linear code over $\mathbb{F}_2$ and let a vector $s$ be chosen uniformly at random in $\mathbb{F}_2^{n-k}$ and an integer $r < n$. Find a vector $x \in B(n, r)$ such that $a(x) = H \cdot x^\top = s$.

When $n = n(\lambda)$, $k = k(\lambda)$, $r = r(\lambda)$, we say that the SD problem $(H, s, r)$ is hard, if the success probability of any PPT algorithm in solving the problem is at most $\text{negl}(\lambda)$.

In our security reduction, the following variant of the left-over hash lemma for matrix multiplication over $\mathbb{F}_2$ is used.

**Lemma 1** (left-over hash lemma [12]). Let $\mathcal{D}$ be a distribution over $\mathbb{F}_2^n$ with min-entropy $e$. For $\varepsilon > 0$ and $n - k \leq e - 2 \log(1/e) - O(1)$, $(H, H \cdot x^\top)$. $H \sim^* \mathbb{F}_2^{(n-k)\times n}$ and $x \in \mathbb{F}_2^n$ is drawn from distribution $\mathcal{D}$, the statistical distance between the distribution of $(H, H \cdot x^\top)$ and the uniform distribution over $\mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{n-k}$ is at most $2^{-\varepsilon}$.

In particular, if $r < n$ is an integer such that $n - k \leq \log \left(\binom{n}{r}\right) - 2\lambda - O(1)$ and $\mathcal{D}$ is the uniform distribution over $B(n, r)$ (i.e., $\mathcal{D}$ has min-entropy $\log \left(\binom{n}{r}\right)$), then the statistical distance between the distribution of $(H, H \cdot x^\top)$ and the uniform distribution over $\mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{n-k}$ is at most $2^{-\varepsilon}$.

2.3. Stern’s Protocol. The identification protocols are very important for enhancing the security of network services and smart cards, where the zero-knowledge (ZK) protocol is a convincing primitive to suggest that a protocol can be reused without any security loss.

An efficient code-based ZK protocol was first proposed by Stern [28] as shown in Figure 1 (we give a modification with employing hamming metric), which is a 3-pass prover-verifier protocol. The prover $\mathcal{P}$ makes a zero-knowledge proof to the verifier $\mathcal{V}$ solving a SD instance $(H, s, r)$ on a low density weight secret $z$. Let $\delta$ be the set of all permutations that keep the propositions in Section 3.1. Hence, the protocol suffers a distinguishable issue since a verifier with the knowledge of $z$ is able to check whether a given witness $z$ was actually involved during the protocol or not. Function $h$ is modelled as a random oracle: $h: \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$. Note that the prover and the verifier in the protocol with a cheating probability $2/3$ are assumed to be honest, in such a way that, the completeness, soundness, and zero knowledge can be thus achieved [29].

Given the public information $(H, s, r) \in \mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{n-k} \times \mathbb{N}$, the prover $\mathcal{P}$ needs to prove the knowledge of $z \in B(n, r)$ such that $H \cdot z^\top = s$.

2.4. Group Signature with Verifier-Local Revocation (VLR-GS). Generally, a VLR-GS scheme consists of the following three algorithms [6]:

1. KeyGen($\lambda, N$): on inputting a security parameter $\lambda$ and group size $N$, this algorithm outputs a group public key $gpk$, a vector of user secret keys...
1. **(Commitment step):** $P$ computes commitments as follows:

\[
c_1 = h(n(HuT)), c_2 = h(n(a)), c_3 = h(n(z + u))
\]

and sends $c_1, c_2, c_3$ to $v$.

2. **[Challenge step]:** $v$ sends $c_1 \leftarrow \{1, 2, 3\}$.

3. **[Response step]:** there are three cases for $v$:

- $v = 1$: $P$ responds: $n(z), n(a)$
- $v = 2$: $P$ responds: $n, z + u$
- $v = 3$: $P$ responds: $n, u$

4. **[Verification step]:** there are three cases for $v$:

- $v = 1$: $v$ checks: $c_2, c_3$
- $v = 2$: $v$ checks: $c_1, c_2$
- $v = 3$: $v$ checks: $c_1, c_2$ and $w(n(z)) = r$.

5. **[Final step]:** $v$ outputs: “Accept”

   If all checks passed; “\(\perp\)” otherwise.

**Figure 1:** Stern’s code-based ZK protocol.

\[
gsk = \langle gsk[0], gsk[1], \ldots, gsk[N - 1] \rangle, \text{ and a vector of users’ revocation tokens}\]
\[
grt = \langle grt[0], grt[1], \ldots, grt[N - 1] \rangle.
\]

(2) **Selfless Anonymity.** The goal of an adversary $A$ in the selfless anonymity game (i.e., Figure 2) is to determine that a signature is generated by which secret key from two adaptively chosen keys. The challenger randomly chooses a bit $b \in \{0, 1\}$ and the adversary $A$ returns a guess $\hat{b}$ on the value. More precisely, $A$ submits two challenge users $d_0$ and $d_1$, and the challenger issues a signature under $d_i$.

In the experiment, the sets $\text{HU}$ and $\text{CU}$ consist of the honest users and corrupted users, respectively. $\sigma^{\text{Sign}}(d, M)$ returns $\text{Sign}(gpk, gsk[d], M)$ and adds $d$ into $S(M)$ if user $d \in \text{HU}$, where the set $S(M)$ is the list of users whose signatures on message $M$ exists. $\sigma^{\text{Corrupt}}(d)$ provides a user’s secret key $gsk[d]$ and moves $d$ to $CU$ if user $d \in \text{HU}$. $\sigma^{\text{Revoc}}(d)$ provides a user’s revocation token $grt[d]$ if user $d \in \text{HU}$.

The adversary $A$ may query an oracle with a group public key gpk at any point during its execution, with the exception that it is not allowed to make any corruption or revocation query for user $d_0$ or $d_1$ in the second query stage. The adversary $A$ wins the game when $\hat{b} = b$, and thus the probability of this event is defined as $\Pr[\mathbf{Exp}_{\text{org}}^{\text{self-anon}-b}(\lambda)]$.

Now, we define the advantage of $A$ in the experiment as

\[
\text{Adv}_{\text{org}}^{\text{self-anon}^{-b}}(\lambda) = \Pr[\mathbf{Exp}_{\text{org}}^{\text{self-anon}^{-1}}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\text{org}}^{\text{self-anon}^{0}}(\lambda) = 1].
\]

(3) **Traceability.** The goal of an adversary in the traceability game is to forge a signature that cannot be traced to which group user using the implicit tracing algorithm. Finally, the adversary $A$ outputs a message $M^*$, a set of revocation tokens $RL^*$, and a signature $\Sigma^*$.

In the experiment, the sets $\text{HU}$ and $\text{CU}$ consist of the honest users and corrupted users, respectively, and the set $U$ consists of corrupt users which is initially empty. Hence, the adversary $A$ wins the game if (i) $\text{Verify}(gpk, RL^*, \Sigma^*, M^*)$ is valid; (ii) the (implicit) tracing algorithm fails or traces to a user outside of the coalition $U^{RL^*}$; (iii) the signature $\Sigma^*$ is nontrivial, i.e., $A$ did not obtain $\Sigma^*$ by making a signing query on $M^*$. The traceability game is formally defined in Figure 3.

The probability that $A$ wins the game is defined as $\Pr[\mathbf{Exp}_{\text{org}}^{\text{tr}}(\lambda, N)]$. We define the advantage of $A$ as

\[
\text{Adv}_{\text{org}}^{\text{tr}}(\lambda, N) = \Pr[\mathbf{Exp}_{\text{org}}^{\text{tr}}(\lambda) = 1].
\]
Experiment $\text{Exp}_{\text{CS}}^N (\lambda, N)$
1. $(gpk, gsk, gtr) \leftarrow \text{KeyGen}(\lambda, N), U = \emptyset.$
2. $(M, \Sigma) \leftarrow A (gpk, gtr[d], G^{\text{sig}}, G^{\text{group}}), U \leftarrow (d) \cup U.$
3. If Verify $(gpk, RL^*, \Sigma^*, M^*) = \text{invalid},$ return 0.
4. If $\exists \not\equiv RL^* \cup U,
   (i, M) \leftarrow A (gpk, gtr : G^{\text{sig}}),$ return 1.
5. Else return 0.

Figure 3: The traceability notion.

3. The Underlying Zero-Knowledge Interactive System

3.1. Some Specific Sets. Before describing the protocol, we introduce several supporting sets of vectors and permutations that will be extensively used throughout this work, which is inspired by Langlois et al. [7]. Let $l$ be a positive integer and $N = 2^l$, and we have

1. (1) Secret$(d)$: the set of all $x = (x_0, x_1, \ldots, x_l \in \mathbb{F}_2^{(2l+1)n})$ that includes $2l + 1$ blocks with size $n$ in $\mathbb{F}_2$. For any user $d = (d[1], \ldots, d[l]) \in \{0, 1\}^l$, the $x_d$ has the following $l$ zero blocks $0^l$: $x_1^d, \ldots, x_l^d[0], x_l^d[1]$.
2. When given $x = (x_0, x_1, \ldots, x_l \in \mathbb{F}_2^{(2l+1)n})$, we define the following two permutations of $x$:
   a. The set $\mathcal{S'}$ of all permutations that keep the following propositions: if $x_i \in \mathcal{S'}$, then $\pi(x) = (\pi_0(x_0), \pi_1(x_1), \ldots, \pi_l(x_l))$, where each permutation $\pi_i$ can transform any word $x_i^b$ for all $b \in \{0, 1\}$ and $i \in [l]$ with same weight that leak any information about their support (in fact, the operation in $[29]$ meets the above proposition: for any $x, \pi(x) \in \mathbb{F}_2^n$, such that $w(x) = w(\pi(x))$, it is possible to find a $n \times n$ invertible matrix $P \in \mathbb{F}_2^{n \times n}$ such that $\pi(x) = xP$).
   b. The set $\mathcal{T'}$ is $\mathcal{T'} = \{T_e \mid e \in \{0, 1\}^l\}$, where for $e = e[1], \ldots, e[l]$, $T_e \in \mathcal{T'}$ is a permutation: $T_e(x) = (x_0, x_1^{e[1]}, \ldots, x_l^{e[1]}, \ldots, x_l^{e[l]}$).

When given $d, e \in \{0, 1\}^l, \pi \in \mathcal{S'}$, and $x \in \mathbb{F}_2^{(2l+1)n}$, it can be checked that

$x \in \text{Secret}(d) \iff \pi(x) \in \text{Secret}(d)$

$\Rightarrow T_e \circ \pi(x) \in \text{Secret}(d \oplus e).$

Therefore, in the following ZK protocol, to prove that $x \in \text{Secret}(d)$ for some $d \in \{0, 1\}^l$ and $A \cdot x' = u$, one can instead prove that for $\forall \pi \in \mathcal{S}', e \in \{0, 1\}^l$, it satisfies

\[
\begin{cases}
T_e \circ \pi(x) \in \text{Secret}(d \oplus e), \\
A \cdot x' = u.
\end{cases}
\]

3.2. The Interactive Protocol. The interactive protocol generally follows Stern’s protocol whose security is reduced to the SD problem and makes a zero-knowledge proof to a verifier $\mathcal{V}$ on a small weight secret $z$ with solving a SD instance $(A, u, r)$. The protocol needs to be repeated $t = r (\log N)$ times to reduce the soundness error negligibly small. The protocol is summarized as follows:

(i) The public parameters consist of a matrix $A = [A_0, A_1, \ldots, A_l] \in \mathbb{F}_2^{(2l+1)n}$, and a vector $u \in \mathbb{F}_2^{(2l+1)}$ and $r \in \mathbb{N}$.
(ii) The witness of a prover includes a vector $x = (x_0, x_1, \ldots, x_l) \in \text{Secret}(d)$ for some secret $d \in \{0, 1\}^l$. The additional input of a verifier is a set $RL = \{(u_1)_{i \in [0, \ldots, N-1]} \in \mathbb{F}_2^{k}$ whose cardinality is at most $N - 1$.
(iii) The goal of the prover is to let the verifier convince the following issues with zero knowledge that 1) $A \cdot x' = u$ and $x \in \text{Secret}(d)$, while $d$ is kept secret; 2) $A_0 \cdot x_0' \not\equiv RL$.

3.2.1. Description of the Protocol. Let COM: $\{0, 1\}^*$ --- $\{0, 1\}^*$ be a collision-resistant hash function to be modelled as a random oracle. The prover’s witness is

$z = (z_0, z_1, \ldots, z_l) \in \mathbb{F}_2^{(2l+1)n}$, for which we employ an additional commitment $e_0$ to enable the revocation mechanism. The prover and the verifier interact as follows.

1. Commitment: the prover samples a string $e \leftarrow \{0, 1\}^l$, a permutation $\pi \leftarrow \mathcal{S'}$, and a vector $r = (r_0, r_1, \ldots, r_l) \leftarrow \mathbb{F}_2^{(2l+1)n}$. Then, it sends the commitment $\text{CMT} = (e_0, c_1, c_2, c_3)$ to the verifier, where

\[
\begin{align*}
\begin{cases}
c_0 &= \text{COM}(e, \pi, A_0 \cdot z_0), \\
c_1 &= \text{COM}(e, \pi, A \cdot z_0), \\
c_2 &= \text{COM}(T_e \circ \pi(r)), \\
c_3 &= \text{COM}(T_e \circ \pi(z + r)).
\end{cases}
\end{align*}
\]

2. Challenge: the verifier sends $ch \leftarrow \{1, 2, 3\}$ to the prover.

3. Response: based on the challenge, the prover computes RSP as

\[
\begin{align*}
\text{Case ch = 1: } RSP &= (d \oplus e, T_e \circ \pi(z), T_e \circ \pi(r)), \\
\text{Case ch = 2: } RSP &= (e, \pi, z + r), \\
\text{Case ch = 3: } RSP &= (e, \pi, r).
\end{align*}
\]

4. Verification: receiving RSP, the verifier outputs Valid if and only if the following three conditions hold. Otherwise, he outputs Invalid. The conditions are as follows:

(i) Case ch = 1: parses RSP and checks
\[
\begin{align*}
T_e \circ \pi(z) & \in \text{Secret}(d\text{w}e), \\
c_2 & = \text{COM}(T_e \circ \pi(r)), \\
c_3 & = \text{COM}(T_e \circ \pi(z) + T_e \circ \pi(r)).
\end{align*}
\]

(ii) Case \( ch = 2 \): parses RSP and checks for \( \forall u \in \text{RL} \):
\[
\begin{align*}
c_0 & \neq \text{COM}(e, \pi, A_0 \cdot (x_0 + r_0)^t - u), \\
c_1 & = \text{COM}(e, \pi, A \cdot (z + r)^t - u), \\
c_2 & = \text{COM}(T_e \circ \pi(r)).
\end{align*}
\]

(iii) Case \( ch = 3 \): parses RSP and checks:
\[
\begin{align*}
c_0 & = \text{COM}(e, \pi, A_0 \cdot r_0^t), \\
c_1 & = \text{COM}(e, \pi, A \cdot r^t), \\
c_2 & = \text{COM}(T_e \circ \pi(r)).
\end{align*}
\]

3.2.2. Analysis of the Protocol. We use the following theorem to summarize some properties of the interactive protocol.

**Theorem 1** (see [7]). If \( \text{COM} \) is a collision-resistant hash function, modelled as a random oracle, then the interactive protocol described in Section 3.2.1 is a ZK protocol of knowledge with perfect completeness, soundness error 2/3, and communication cost \( 1 \cdot \theta(n) \).

In particular, there exists an efficient knowledge extractor that, on inputting a commitment CMT and 3 valid responses (RSP\((1)\), RSP\((2)\), RSP\((3)\)) corresponding to all 3 possible values of the challenge \( ch \), outputs a vector \( y = (y_0||y_1||\ldots||y_9) \in F_2^{20+n} \) satisfying \( A \cdot x^t = u \), where \( y \in \text{Secret}(d) \) for some \( d \in [0,1]^l \), and \( A_0, y_9 \notin \text{RL} \). It can be seen that the given interactive protocol is perfectly complete, i.e., if the prover possesses a valid witness \( z \), the verifier always outputs Valid according to this protocol. Indeed, given \( z \) satisfying \( z \in \text{Secret}(d) \) and \( A \cdot z^t = u \), as discussed above, prover can always obtain \( z \) satisfying \( A \cdot z^t = u \) and \( T_e \circ \pi(x) \in \text{Secret}(d\text{w}e) \). As a result, prover should always pass verifier’s verification in the case \( ch = 1 \) and \( ch = 2 \), and in the case \( ch = 3 \), it suffices to note that verifier simply checks for honest computations of \( c_1 \) and \( c_2 \).

Besides, the interactive protocol is statistically zero knowledge if \( \text{COM} \) is modelled as a random oracle [12].

4. A Code-Based VLR-GS

In this section, we will describe and analyze our code-based VLR-GS scheme. First, we need to introduce an important tool for our scheme: \textit{trapdoor matrix}, which permits to find a small weight preimage of the previous random syndrome to which a fixed syndrome is added.

**Definition 3** (trapdoor matrix [23]). A trapdoor matrix family is a couple of polynomial algorithms (\( \text{TrapGen}, \text{SampleS} \)) such that

(i) \( \text{TrapGen}(\lambda) \): outputs a pair \((A, \text{trk}) \in F_2^{(n-k)n} \times F_2^k \) according to the security parameter \( \lambda \), where the \( F_2^k \) is the value range of trk.

(ii) \( \text{SampleS}(A, \text{trk}, u, r) \): outputs with a non-negligible probability, some \( x \in B(n, r) \) such as \( A \cdot x^t = u \) assuming \((A, \text{trk}) \)—\( \text{TrapGen}(\lambda) \), \( u \in F_2^n \) and \( r \in \mathbb{N} \). Notice that \( x \) appears randomly from \( B(n, r) \), and it returns \( \bot \) if no solution is found.

(iii) (Correctness): for all \((A, \text{trk})\) — \( \text{TrapGen}(\lambda) \) and all vectors \( u \in F_2^n \), we have \( A(\text{SampleS}(A, \text{trk}, u, r)) = u^t \).

(iv) (One-wayness): for all PPT adversary \( \mathcal{A} \), we have a negligible probability:
\[
\text{Pr}[A \cdot x^t = u, x \in B(n, r)|(A, \text{trk}) \leftarrow \text{TrapGen}(\lambda); \mathcal{A} \leftarrow \mathcal{A}(\lambda, A, u, r)] 
\]

Hence, we say that \( A \) is a trapdoor matrix and trk is a trapdoor key if \((A, \text{trk})\) was generated by \( \text{TrapGen} \).

4.1. Description of the Scheme

4.1.1. Key Generation. On input the security parameter \( \lambda \), an expected number of group users \( N = 2^l \in \text{poly}(\lambda) \) and a positive integer \( l \). The key generation algorithm \( \text{KeyGen}(\lambda, N) \) works as follows:

(1) Select parameters \( n = n(\lambda), k = k(\lambda), r = r(\lambda) \) for the syndrome decoding problem, such that \( n - k \leq \log(n) - 2l - \Theta(1) \).

(2) Run \( \text{TrapGen}(\lambda) \) to get a trapdoor matrix \( A_0 \in F_2^{(n-k)n} \) and a trapdoor key trk.

(3) Sample \( u \in F_2^n \) and \( A_b^* \leftarrow F_2^{(n-k)n} \) for all \( b \in [0,1], i \in [l] \). Define the matrix \( A = [A_0 | A_1 | \ldots | A_l] \in F_2^{(n-k)(2l+1)n} \).

(4) For a group user with index \( d = (d_1, \ldots, d[l]) \in [0,1]^l \), do the following:

(a) Sample vectors \( x^{[d]}_0, \ldots, x^{[d]}_l \leftarrow B(n, r) \) and compute \( z = \sum_{i=1}^l A_i^{[d]} \cdot x^{[d]}_i \). Then, sample vector \( x_0 \in F_2^n \) with \( x_0 = \text{SampleS}(A_0, \text{trk}, u - z, r) \). Let \( x^{[d]}_1, \ldots, x^{[d]}_{l-1} \) be zero vectors \( \Theta^n \), and define \( x^{[d]}_l = (x_0 \| x_0 \| x_1 \| \ldots \| x_0 \| x_l) \in F_2^{(2l+1)n} \); if \( w(x^{[d]}_l) = r \), then go to step (b); else, repeat step (a).

(b) Let the secret key of the group manager be gmsk = trk, the user secret key be gsk\([d]\) = \( x^{[d]} \), and the revocation token be grt\([d]\) = \( A_0 \cdot x^{[d]} \) where

(5) The algorithm outputs gmsk = trk and (gpk, gsk, grt) where
follows:

\[ gpk = (A, u), \]
\[ gsk = (gsk[0], gsk[1], \ldots, gsk[N - 1]), \quad (12) \]
\[ grt = (grt[0], grt[1], \ldots, grt[N - 1]). \]

4.1.2. Signing Algorithm. Let \( \mathcal{H} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^t \) be a hash function, modelled as a random oracle, where \( t = r(\log n) \) is the repeat number of the ZK protocol described in Section 3.2.1. Given \( gpk = (A, u) \), we use the secret key \( gsk[d] = x \in \text{Secret}(d) \) to sign a message \( M \in \{0, 1\}^* \), and the user runs \( \text{Sign}(gpk, gsk[d], M) \) as follows:

1. Generate a proof that the user is a certified group member and has not been revoked; this is done by repeating the ZK protocol with \( t \) times to achieve a negligible soundness error. With the public parameter \( (A, u) \) and prover's witness \( x \), we make it noninteractive with the Fiat–Shamir heuristic as a triple:

\[ \Pi = \left( \{ \text{CMT}^{(i)} \}_{i=1}^t, \text{CH}, \{ \text{RSP}^{(i)} \}_{i=1}^t \right), \quad (13) \]

where \( \text{CH} \in \{1, 2, 3\}^t \) as \( \text{CH} = \{ \text{ch}^{(i)} \}_{i=1}^t = \mathcal{H}(M, \{ \text{CMT}^{(i)} \}_{i=1}^t) \).

2. Output the group signature:

\[ \Sigma = (M, \Pi). \quad (14) \]

4.1.3. Verification Algorithm. On inputting \( gpk = (A, u) \), a set of tokens \( RL = \{ \{ u \} \}_{i=0, \ldots, N-1} \subset \mathbb{F}_2^{n \times k} \) whose cardinality is at most \( N - 1 \), a message \( M \in \{0, 1\}^* \), and a purported group signature \( \Sigma \) on \( M \), the verifier runs the deterministic algorithm \( \text{Verify}(gpk, RL, \Sigma, M) \) as follows:

1. Parse the group signature \( \Sigma \) as \( (M, \Pi) \) as in formula (14).

2. Return Invalid if

\[ \left( \text{ch}^{(1)}, \ldots, \text{ch}^{(t)} \right) \neq \mathcal{H}(M, \text{CMT}^{(1)}, \ldots, \text{CMT}^{(t)}). \quad (15) \]

3. For \( i = 1, \ldots, t \), run the verification of the ZK protocol to check the validity of \( \text{RSP}^{(i)} \) with respect to \( \text{CMT}^{(i)} \) and \( \text{ch}^{(i)} \). Output Invalid and terminate, if any of the verification conditions does not hold.

4. Output Valid.

Remark 2. According to the above key generation algorithm, we can get some observations as follows:

1. According to Lemma 1, the distribution of \( A_0 \) is statistically close to uniform over \( \mathbb{F}_2^{(n-k) \times n} \). Hence, the distribution of \( gpk \) is statistically close to uniform over \( \mathbb{F}_2^{(n-k) \times (2^{2r+1} n) \times 2^{n-k}} \).

2. The secret key \( x^{(d)} \) of a group user indexed by \( d \) satisfies \( A \cdot x^{(d)} = u \) and \( x^{(d)} \in \text{Secret}(d) \).

3. The distribution of revocation token \( grt[d] \) is statistically close to uniform over \( \mathbb{F}_2^{(n-k)} \), and the tokens of two different group users should be different. In each rare event of conflict, the algorithm simply resamples the key and token for the user indexed by \( d_2 \).

4.2. Security and Efficiency Analysis

4.2.1. Security. The correctness, selfless anonymity, and traceability of our VLR-GS are stated in Theorems 2–4, respectively.

Theorem 2. The VLR-GS scheme described in Section 4.1 is correct with an overwhelming probability.

Proof. According to the correctness requirement, it is necessary to prove that for all \( gpk = (A, u) \),

\[ \text{gsk} = (gsk[d]_{N-1}^{d}), \quad \text{grt} = (grt[d]_{N-1}^{d}) \]

generated by KeyGen(\( \lambda, N \)), \( d \in \{0, 1\}^t \), and all \( M \in \{0, 1\}^* \), we have

\[ \text{Verify}(gpk, RL, \text{Sign}(gpk, gsk[d], M), M) = \text{Valid} \quad (16) \]

We can observe that, with overwhelming probability, the honest signer index \( d \in \{0, 1\}^t \) can obtain a valid witness \( x \) to be used in the underlying argument system. Then, thanks to the perfect completeness of the latter protocol, the II satisfy formula (5) is a valid triple and the algorithm \( \text{Verify}(gpk, RL, \Sigma, M) \) always outputs Valid for any message \( M \in \{0, 1\}^* \).

Suppose that the verification algorithm outputs Valid, and we have \( grt[d] \notin RL \). If there exists an index \( i \) such that \( grt[d] = u_i \), i.e., a group user indexed by \( i \) with the secret key \( x^{(d)} \) satisfies \( A_0 \cdot x^{(d)} = u_i \) and \( x^{(d)} \in \text{Secret}(d) \), the verification algorithm always outputs Valid since it cannot pass the verification of the interactive protocol described in Section 3.2.1. Finally, the verification algorithm outputs Valid with an overwhelming probability when \( grt[d] \notin RL \).

Theorem 3. The proposed code-based VLR-GS scheme described in Section 4.1 is selfless-anonymous under the SD assumption in the random oracle model.

Proof. Let \( \mathcal{A} \) be any PPT adversary that aims to break the selfless anonymity security of the scheme. We define a sequence of hybrid games \( G_0^{(b)}, G_1^{(b)}, G_2^{(b)}, \) and \( G_3 \), for \( b \in \{0, 1\} \), where

- Game \( G_0^{(b)} \): the original selfless anonymity game in which the bit chosen by the challenger is \( b \).
- Game \( G_1^{(b)} \): a game similar to \( G_0^{(b)} \), except the game \( G_1^{(b)} \) performs signing algorithm in an honest manner, and generates a legitimate signature.
Game $G_2^{(b)}$: a game similar to $G_1^{(b)}$, except the game $G_2^{(b)}$ replaces the token $grt[d_b]$ with a value selected uniformly random over $F_2^{m-k}$.

Game $G_3$: a game executes a proper SD problem instance.

We continue to prove that each of the two games is indistinguishable based on the zero-knowledge property of the underlying argument and the hardness of the SD problem. In the end, the selfless anonymity of our scheme then follows from the fact that game $G_3$ is independent of the bit $b$.

**Game $G_0^{(b)}$:** The game runs as follows.

1. Run KeyGen($\lambda$, $N$) to obtain $(gpk, gsk, grt)$, set $RL\leftarrow\emptyset$, Corrupted $\leftarrow\emptyset$, and give gpk to the adversary $A$.
2. Based on the queries issued by $A$, we have
   - (i) If $A$ queries the signature on any message $M$ and user $d$, return $\Sigma = \text{Sign}(gpk, gsk[d], M)$.
   - (ii) If $A$ queries the corruption of user $d$, set $\text{Corrupted} := \text{Corrupted} \cup \{d\}$ and return $gsk[d]$.
   - (iii) If $A$ queries the revocation of user $d$, set $RL = RL \cup \{d\}$ and return $grt[d]$.
3. $A$ outputs a message $M^*$ and $d_0, d_1$ such that $d_0, d_1 \notin \text{Corrupted}$, and $grt[d_0], grt[d_1] \notin RL$.
4. Generate a signature $\Sigma = \text{Sign}(gpk, gsk[d_0], M^*) = (M^*, \Pi^*)$, and return $\Sigma$ to $A$.
5. $A$ can still make queries as before, but not allowed to ask for $gsk[d_0], gsk[d_1], grt[d_0]$ and $grt[d_1]$.
6. Finally, $A$ outputs a bit $b'$.

**Game $G_1^{(b)}$:** This game makes the following modification with respect to Game $G_0^{(b)}$ in Step 4, instead of generating a signature as follows: run the preparation steps of the signing algorithm in an honest manner; then simulate the noninteractive ZK protocol $\Pi^*$. This is done by invoking the simulator of the argument in Theorem 1 for each repetition and programming the random oracle $F$ accordingly. Since the underlying argument is statistically zero knowledge, the distribution of $\Pi^*$ is statistically close to that of the legitimate $\Pi$. Output $\Sigma^* = (M^*, \Pi^*)$.

It can be seen that $\Sigma^*$ is statistically close to the signature $\Sigma$ outputted by Step 4 in Game $G_0^{(b)}$. Hence, we can conclude that Game $G_1^{(b)}$ and Game $G_0^{(b)}$ are statistically indistinguishable.

**Game $G_2^{(b)}$:** In this game, we introduce the following modification with respect to Game $G_1^{(b)}$. In $G_1^{(b)}$, the token $grt[d_b]$ is unknown to $A$, since it is statistically close to uniform over $F_2^{m-k}$ and $grt[d_b] = A_0 \cdot x_0^{(b)}$ with $(A_0, x_0^{(b)}) \in F_2^{-k} \times F_2^{-k}$. Here, we replace $x_0^{(b)}$ as $s_0 \cdot r^* F_2^{m-k}$ and compute $grt[d_b] = A_0 \cdot s_0$. The remaining of the game is same as in Game $G_1$. Hence, we can conclude that Game $G_2^{(b)}$ and Game $G_1^{(b)}$ are statistically indistinguishable.

**Game $G_3$:** Note that in previous games, the pair $(grt[d_b], s_b)$ is a right SD instance, and hence its distribution is computationally close to the uniform distribution over $F_2^{m-k} \times F_2^{k}$ under the SD assumption. In the game, we let $grt[d_b]$ truly be uniform by sampling $w \leftarrow F_2^{m-k}$ and setting $grt[d_b] \leftarrow w$. The remaining of the game is same as in Game $G_2^{(b)}$. Clearly, the advantage of the adversary in this game is 0. Hence, we can conclude that $G_2^{(b)}$ and $G_3$ are computationally indistinguishable if the SD problem is hard.

Finally, based on the combined observations on $G_0^{(b)}, G_1^{(b)}, G_2^{(b)}, G_3$ for the bit $b \in \{0, 1\}$, we have

$$G_0^{(b)} = G_1^{(b)} = G_2^{(b)} \approx G_3,$$

(17)

where “$\approx$” means statistically indistinguishable and “$\approx$” means computationally indistinguishable. The result shows that the advantage of an adversary winning in game $G_0^{(b)}$ for $b \in \{0, 1\}$ is negligible. This concludes the proof.

**Theorem 4.** Suppose there is a PPT traceability adversary $A$ against our VLR-GS scheme with a success probability $\xi$ in the random oracle model; then, there is an algorithm $F$ that solves the SD problem with a success probability polynomially related to $\xi$.

**Proof.** Suppose there is an adversary $A$ that can break the computational binding property of COM employed by the underlying argument with a non-negligible probability; then, we can use $A$ to solve the SD problem. Therefore, without loss of generality, we assume that COM is a collision-resistant hash function. Now, we construct a PPT algorithm $F$ solving the SD problem with a non-negligible probability, which works as follows.

**Challenge:** the algorithm $F$ receives a challenge SD instance, that is, a uniformly random matrix $C = [C_0|C_1|\ldots|C_l] \in F_2^{(n-k) \times (l+1)n}$ and a vector $\mathbf{u} \in F_2^{(n-k) \times (l+1)n}$, and it wins the challenge if it can produce a nonzero vector $\mathbf{x} \in F_2^{(l+1)n}$ such that $w(\mathbf{x}) = r$ and $C \cdot \mathbf{x} = \mathbf{u}$.

**Setup:** $F$ performs the following steps:

1. (1) Sample $\mathbf{z} = (z_0, z_1, \ldots, z_l) \in F_2^{(l+1)n}$, where each coordinate of $\mathbf{z}$ is sampled from $F_2$, with length $n$. If $w(\mathbf{z}) > r$, then repeat the sampling; otherwise, compute $\mathbf{u} = C \cdot \mathbf{z}$.
2. (2) Run TrapGen($\lambda$) with $l$ times and let the outputs be $((P_1, R_1), (P_2, R_2), \ldots, (P_l, R_l))$.
3. (3) Pick a target index $d^* = (d^*[1], \ldots, d^*[l]) \leftarrow \{0, 1\}^l$, and define $A = [A_0|A_0^0|A_1^0|\ldots|A_l^0|A_l^1] \in F_2^{(n-k) \times (2l+1)n}$, where $A_0 = C_0$, and for each $i \in [l]$, $A_i^{d^*[i]} = C_i$ and $A_i^{1-d^*[i]} = P_i$.
4. (4) Define the secret key and revocation token for user $d^*$ as follows:

$$\text{gsk}[d^*] = (x_0, x_1, \ldots, x_l),$$

where for $i \in [l]$: $s_0 = x_0, x_i^{d^*[i]} = x_i, x_i^{1-d^*[i]} = 0^n$, $grt[d^*] = A_0 \cdot x_0$.
5. (5) Generate the secret key and the revocation token for each user $d[b] \neq d^* [b]$, where $d = (d[1], \ldots, d[l]) \in \{0, 1\}^l$, as follows:
(a) Let $d [b]$ ($1 \leq b \leq l$) be the first bit from the left where $d [b] \neq d' [b]$. Since $d \neq d'$, then $b$ exists. It follows that $A_d [b] = A_d' [b] = P$. 

(b) Sample $x_0^d, x_1^d, \ldots, x_{d-1}^d, x_{d+1}^d, \ldots$ 

\[ x_i^d \rightarrow \sim \arrow{B(n, r)} \] 

(c) Sample $x_0^{d[i]} \rightarrow$ Sample's $(R_0, \text{trk, } t^{d[i]}, r)$. 

(d) For each $i \in [l]$, let $x_i^{d[i]} = 0^r$ and $x^{d[i]} = (x_0^d, x_1^d, \ldots, x_{d-1}^d) \in \mathbb{F}_2^{(2l+1)n}$. If each random event that $\omega(x^{d[i]}) > r$ happens, then repeat the sampling. Otherwise, set $\text{gsk}[d] = x^{d[i]}$ and $\text{g} \left[ d \right] = A_0 \cdot x_i^d \in \mathbb{F}_2^n$. 

Note that by the construction, the distribution of $(\text{gpk}, \text{gsk}, \text{g})$ is statistically close to that of the real scheme, and the choice of $d^*$ is hidden from the adversary. Hence, the algorithm $F$ gives $(\text{gpk}, \text{g})$ to $\mathcal{A}$. 

Queries: the algorithm $F$ answers the queries of $\mathcal{A}$ as follows: 

(i) Corruption queries: at the beginning, the corruption set $\mathcal{U}$ is set to be empty. If $\mathcal{A}$ queries the secret key of any user $d \in \{0, 1\}$, then $F$ adds $d$ to the set $\mathcal{U}$ and returns $\text{gsk}[d]$.

(ii) Signatures queries: if $\mathcal{A}$ queries a signature of user $d$ on an arbitrary message $M$, then $F$ returns $\Pi = \text{Sign}(\text{gpk}, \text{gsk}[d], M)$. The queries to the random oracles $\mathcal{H}$ and $\mathcal{G}$ are handled by consistently returning uniformly random values in the respective ranges. For each $t \leq q_{3r}$, let $r_t$ denote the answer to the $t$-th query.

Forgery: eventually, $\mathcal{A}$ outputs a message $M^*$, a set of tokens $RL^*$, and a nontrivial forged signature 

\[ \Pi^* = \left\{ \left\{ \text{CMT}[0] \right\}_{i=1}^l, \left\{ \text{ch}[i] \right\}_{i=1}^l, \left\{ \text{RSP}[0] \right\}_{i=1}^l, M^* \right\}. \] 

such that $\text{Verify}(\text{gpk}, RL^*, \Pi^*, M^*) = \text{Valid}$, and the implicit tracing algorithm fails or traces to a user outside of the coalition $U/RL^*$. The algorithm $F$ exploits the forgery as follows.

(1) One can argue that $\mathcal{A}$ must have queried $\mathcal{G}$ on inputs $(M^*, \{ \text{CMT}[0] \}_{i=1}^l)$; otherwise, the probability behaves as

\[ \text{Pr}[(\text{ch}^{(1)}_1, \ldots, \text{ch}^{(1)}_l)] = F(M^*, \{ \text{CMT}[0] \}_{i=1}^l) \leq 3^{-t}, \] 

for the choice of $d$ being completely hidden from $\mathcal{A}$'s view. Therefore, the probability is at least $\xi - 3^{-t}$, and there certainly exists $t^* \leq q_{3r}$ such that $(M^*, \{ \text{CMT}[0] \}_{i=1}^l)$ is the $t^*$-th oracle query.

(2) $F$ picks $t^*$ as the target forking point and replays $\mathcal{A}$ many times with the same random tape and inputs as in the original run. In each rerun, for the first $t^* - 1$ queries, $\mathcal{A}$ is given the same answers $r_{1}, \ldots, r_{t^*-1}$ as in the initial run; however, from the $t^*$-th query onwards, $F$ replies with random values $r_{t^*}, \ldots, r_{q_{3r}} \rightarrow \sim \{1, 2, 3\}$. The improved forking lemma of Pointcheval and Vaudenay ([30], Lemma 7) implies that with high probability, there is at least 1/2, the algorithm $F$ can obtain a 3-fork involving the tuple $(M^*, \{ \text{CMT}[0] \}_{i=1}^l)$ after less than $(32 \cdot q_{3r}/(\xi - 3^{-t}))$ replays of $\mathcal{A}$. Now, let the answers of $F$ with respect to the 3-fork branches be

\[ \left\{ \begin{array}{l}
 r_{t^*}^{(1)} = (\text{ch}_1^{(1)}, \ldots, \text{ch}_1^{(1)}), \\
 r_{t^*}^{(2)} = (\text{ch}_2^{(1)}, \ldots, \text{ch}_2^{(1)}), \\
 r_{t^*}^{(3)} = (\text{ch}_3^{(1)}, \ldots, \text{ch}_3^{(1)}).
 \end{array} \right. \] 

Intuitively, we can get that $\text{Pr}[(3 \in [1]): (\text{ch}_1^{(1)}, \text{ch}_2^{(1)}, \text{ch}_3^{(1)}) = (1, 2, 3)] = 1 - (7/9)^t$. With the index $i$, one parses the 3 forgeries corresponding to the fork branches to obtain $(\text{RSP}^{(1)}, \text{RSP}^{(2), \ldots, \text{RSP}^{(3))}$ and make them be 3 valid responses in relation with 3 different challenges for the same CMT$_r$. Since COM is a collision-resistant hash function, we can invoke Theorem 1 to extract $y = (y_0^{(1)}, y_1^{(1)}, \ldots, y_n^{(1)}) \in \mathbb{F}_2^{(2l+1)n}$ by $y_i^{(1)} = \text{Sign}(\Pi, r_{t^*}^{(i)}, \text{r}_{t^*}^{(i)}, y_{t^*}^{(i)})$ obtained from removing the zero blocks $y_i^{(1)}, \ldots, y_i^{(1)}$ from $y$. Note that by the construction, one has $C \cdot y^{(1)} = A \cdot y^* = u$ and $y^* \in B((l + 1)n, r)$. In other words, $y^*$ is a valid solution to the challenge SD instance $(C, u, r)$.

Finally, we can conclude that if $\mathcal{A}$ has a success probability $\xi$ with running time $T$ in attacking the traceability of our group signature scheme, then $F$ has the success probability of at least $(1/2 \cdot 1/(N - (2/3)^{k})) \cdot (1 - (7/9)^t)$ with running time at most $32 \cdot T = q_{3r}/(\xi - 3^{-t}) + \text{poly}(n, N)$. This concludes the proof of the traceability property. \hfill $\square$

4.2.2. Efficiency. The algorithms described in Section 4.1 can be implemented in polynomial time by selecting appropriate parameters. Asymptotically, for any $n, k$, code and a group with $N = 2^l$ group members, the public key size is $\text{log}(N) \cdot O(n^2)$ over $\mathbb{F}_2$, and the group signature size is $t \text{log}(N) \cdot O(n) + 4\lambda$ over $\mathbb{F}_2$, where $t = r = \text{log}(n)$. This is the number of repetitions of the protocol described in Section 3.2.1.

In addition, we have made a comparative analysis of the latest research studies of revocable group signature schemes in Table 1. In this table, references [18, 19] are two VLR-GS
frameworks with the traditional “sign-and-encrypt” paradigm, which can achieve a high level of anonymity: (almost-) full anonymity. But, the code-based instantiation scheme seems not readily available, as code-based signatures for which there are efficient zero-knowledge proofs of knowledge of message/signature pairs are not known to date. Scheme [27] is a lattice-based VLR-GS scheme; like our scheme, it can resist quantum computer attacks. Scheme [27] achieves full anonymity. However, it cannot be realized in code-based group signature scheme temporarily. The scheme [12] introduced the first provably secure group signature scheme from code-based assumptions in a static group. References [23, 31] are the other two code-based group signature schemes, and they allow to dynamically add new members (weakly dynamic) but cannot realize the membership revocation. In summary, our scheme is the first code-based group signature scheme that realizes membership revocation, but unfortunately, it does not serve member registration. Next, we will consider further constructing a fully dynamic group signature system simultaneously supporting dynamic user enrolments and user revocations.

4.3. Implementation Results. Our code-based VLR group signature scheme can be implemented in polynomial time by selecting parameters carefully. For a \([n, k]\) code, a group with \(N = 2^l\) group members, asymptotically, the public key mainly consists of \((\mathbf{A}, \mathbf{u}) \in \mathbb{F}_2^{(n-k) \times (2^{l+1}n)} \times \mathbb{F}_2^{n-k}\), and the group public key size is \(l \cdot O(n^2) = \log(N) \cdot O(n^2)\) over \(\mathbb{F}_2\). For the group signature size, the commitment CMT achieves the bit size with \(4\lambda\) (the security parameter \(\lambda\) should be greater than 80), and for each of the cases \(c = 1, c = 2,\) and \(c = 3,\) the size of response RSP is \(4nl + 2n + I\), and hence the group signature size is \(tl \cdot O(n) + 4\lambda = t \log(N) \cdot O(n) + 4\lambda\) over \(\mathbb{F}_2\), where \(t\) is the repetition number of the protocol described in Section 3.2.1. The running time of revocation check (i.e., the check against \(c_0^{(k)}\) when \(c^{(k)} = 2\)) is linear with the number of revoked users.

We present the basic implementation results to demonstrate the feasibility, where the scheme is generic and can be used with any code. The binary Goppa codes are embedded since they constitute well-suited candidates for the instantiating introduced in our scheme. To achieve a 80-bit security level, we set the SD parameters as \((n, k, r) = (2^{11}, 1696, 121)\), so that the distribution of \(u\) is \(2^{80}\) close to the uniform distribution over \(\mathbb{F}_2^{n-k}\) and that the SD problem is intractable with respect to the best known attacks. The protocol is repeated at least 140 times to make the soundness error negligibly small (less than \(1 - 2^{-80}\)). We employ SHA-3 as an instantiation for the used hash functions, where it leads to the public key of size \(8.9 \times 10^6\) bits for \(2^8 = 256\) users and \(1.7 \times 10^7\) bits for \(2^{16} = 65536\) users. And the signature size is \(6.8 \times 10^5\) bits for \(2^8\) users and \(1.3 \times 10^8\) bits for \(2^{16}\) users, with the bit size of any message \(M \in \{0, 1\}^\ast\).

5. Conclusion

In this work, we propose the first code-based group signature scheme with membership revocation, whose security is reduced to the hardness of the syndrome decoding problem. The main idea of our work is to build a new Stern-like interactive identification protocol with member revocation mechanism. Our structure departs from the traditional “sign-and-encrypt-and-prove” group signature paradigm and does not rely on a specific encryption primitive.

To the best of our knowledge, the construction presented here is the first one for code-based group signature scheme that supports member revocation, although it could still be improved for efficiency. In the future work, we will focus on improving the current building in terms of performance (for example, achieving both dynamic registration and efficient revocation of users), efficiency (for example, making the group signature size less dependent or independent of the group size), and CCA anonymity.

Moreover, in terms of practical application, group signature scheme with membership revocation is a useful cryptographic primitive. With the development of the openness of wireless network, the mobile device and application (app) security are increasingly important [32], and authentication and anonymity issues in mobile environments must be concerned. The network communication between mobile users and cloud servers may suffer from various attacks, such as impersonation attack and password guessing attack. So, it is indispensable to establish an authentication and key agreement (AKA) protocol to protect the conversations between the users with mobile devices and remote servers in various application environments. To be practical, a two-factor/multifactor AKA protocol [33, 34] must satisfy the following security goals: (1) truly two-factor/ multifactor security, that is, if an attacker gains any part of the authentication factors, the attacker cannot successfully figure out the remaining factors; (2) anonymity and untraceability, including identity protection and user untraceability; (3) local and secure password update.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Technology</th>
<th>Anonymity</th>
<th>Join</th>
<th>Revocation</th>
<th>Postquantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>Number theory</td>
<td>Full anonymity</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[19]</td>
<td>Number theory</td>
<td>(Almost-) full anonymity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[27]</td>
<td>Lattice</td>
<td>Full anonymity</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[12]</td>
<td>Code</td>
<td>CPA anonymity</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>[31]</td>
<td>Code</td>
<td>Anonymity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[23]</td>
<td>Code</td>
<td>Selfless anonymity</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Ours</td>
<td>Code</td>
<td>Selfless anonymity</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: The performance comparison with related works.
Actually, the unforgeability, anonymity, and other attributes of an AKA protocol just meet the security requirements of a VLR-GS scheme. So, we firmly believe that the application research studies of VLR-GS system in mobile device authentication and two-factor/multifactor authentication schemes are worthy of discussion in the future. Moreover, we can consider using a zero-knowledge proof protocol to ensure the security of user identity, utilizing a key encapsulation mechanism to protect the session-key security in an AKA protocol.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61972156, 61802248, 61632012, U1705264, and 61961146004), the "Chenguang Program" supported by Shanghai Education Development Foundation (61972156, 61802248, 61632012, 61972156), the National Key Research and Development Program of China (2018YFA0704701), and the Fundamental Research Funds for the Central Universities.

References


