Information Sciences 589 (2022) 751–769

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Revisiting group oriented secret sharing schemes

Rui Xu [a], Xu Wang [b], Kirill Morozov [c], Chi Cheng [a,*], Jintai Ding [d]

[a] School of Computer Science, China University of Geosciences (Wuhan), China
[b] School of Computer Science and Technology, University of Science and Technology of China, China
[c] Department of Computer Science and Engineering, University of North Texas, USA
[d] Ding Lab, Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing, China

ARTICLE INFO

Article history:
Received 11 July 2021
Received in revised form 20 October 2021
Accepted 11 December 2021
Available online 6 January 2022

Keywords:
Secret sharing
Group oriented secret sharing
Cryptanalysis
LWE

ABSTRACT

In a (t, n) threshold scheme any t or more shares can reconstruct the secret s, but less than t shares reveal no information about s. However, an unauthenticated adversary can pretend to be the shareholder at the reconstruction stage. If there were more than t honest shareholders, the unauthenticated adversary without valid share can obtain the secret. To deal with this type of attacks, a model of (t, m, n) group oriented secret sharing (GOSS) scheme was proposed by Miao et al. in 2015. Here the group oriented property means that if $m > t$ parties try to reconstruct the secret, they should *all* have the authentic shares in advance. It was claimed by Miao et al. that the group oriented property in their GOSS schemes holds in the information-theoretic sense. In this paper, we revisit two instantiations of (t, m, n) group oriented secret sharing schemes and show that these constructions cannot provide the so-called "group oriented property". Specifically, we develop concrete attacks which allow an unauthenticated adversary with no valid share to participate in the reconstruction phase and obtain the secret provided that there are at least t honest shares presented at the reconstruction phase.

1. Introduction

Threshold secret sharing schemes were independently introduced by Shamir [1] and Blakerly [2]. In a (t, n) threshold scheme, a dealer D can split her secret s into n shares $(s_1, \ldots, s_n)$ and send each share s_i privately to a user (also called participant or shareholder) U_i, for $i = 1, \ldots, n$. The basic security requirement for (unconditionally secure) threshold schemes is that at least t users can reconstruct the original secret s by putting these shares together while less than t shares reveal no information regarding s. In addition to Shamir's polynomial-based construction and Blakerly's geometry-based construction, Asmuth and Bloom [3] also proposed an unconditionally secure threshold scheme based on the Chinese Remainder Theorem (CRT). When referring to *unconditional* security, we imply that it holds in an information-theoretical sense, even against adversaries with unlimited computing resources.

Secret sharing schemes have lots of applications in cryptography such as threshold cryptography [4], secure multi-party computation [5,6] and secure data storage [7], to mention a few. Harn [8] pointed out a limitation in the original model of threshold schemes. Namely, threshold schemes typically assume that at the reconstruction phase all participants are authenticated shareholders. Thus, one cannot prevent an unauthenticated outsider, who has no valid share, from obtaining the

[*] Corresponding author.
E-mail address: chengchi@cug.edu.cn (C. Cheng).

https://doi.org/10.1016/j.ins.2021.12.053
0020-0255/© 2021 Elsevier Inc. All rights reserved.

shared secret s. Let us consider a situation where m participants join the reconstruction phase with $m \geqslant t+1$, while one of the m participants is an outsider adversary who is an unauthenticated user with no valid share. The adversary can still obtain the secret s since she can see $m-1 \geqslant t$ authentic shares during the reconstruction phase. There have been various models of secret sharing which can protect traditional secret sharing under different adversarial environments. For example, verifiable secret sharing [9,10], threshold changeable secret sharing [11,12] and cheater identifiable secret sharing [13]. Although the above secret sharing variants can in fact deal with the problem of unauthenticated outsider adversary, they are designed for more powerful adversaries and are thus heavy in computation and communication costs. Thus, in order to address the issue with unauthenticated users in secret sharing schemes, Harn proposed a notion called secure secret reconstruction scheme. In Harn's secure secret reconstruction scheme [8], an outside adversary with no valid share cannot obtain the secret s, even if there are more than t authentic shares in the reconstruction phase. Unfortunately, Harn's scheme was proved to be insecure [14].

Later, Miao et al. [15,16] proposed a notion called (t,m,n) Group Oriented Secret Sharing (GOSS). GOSS has the group oriented property that the secret can be reconstructed only if all m $(m>t)$ participants provide valid shares at the reconstruction phase. Thus unauthenticated users cannot obtain the secret. Besides protecting the secret from unauthenticated users, secure secret reconstruction scheme or GOSS can also be applied in group authentication schemes. In group authentication schemes, a set of group members can mutually authenticate each other effectively without relying on authenticating in a pairwise manner or sending messages to a central server. Harn [17] constructed a (t,m,n) group authentication scheme based on his secure secret reconstruction protocol. However, as we stated earlier, due to the security vulnerability in Harn's secure secret reconstruction, the proposed (t,m,n) group authentication scheme is insecure [14,18] against impersonation adversary. In fact, due to limited understanding of the security of GOSS, lots of group authentication schemes [17–20] have security loopholes [21,22] in their design.

Hence, in this work we devote ourselves in analyzing the theoretical and practical security of two concrete constructions for (t,m,n) GOSS, one [15] based on Shamir's scheme and the other [16] based on Asmuth-Bloom scheme. We will refer the two GOSS schemes as polynomial-based GOSS and CRT-based GOSS, respectively. Simply speaking, the two (t,m,n) GOSS schemes use the randomized components (RCs) in the reconstruction phase, which are the original shares masked with some random value. Thus, the secret s can only be reconstructed if all RCs are correct. The GOSS scheme is claimed to guarantee the group oriented property in the unconditional security setting. Moreover, the two (t,m,n) GOSS schemes claim that the RC can hide the share in a way that the shareholder can employ her share more than once to construct different RCs without exposing the share. Note that we only consider an outside adversary in this paper for two reasons. Firstly, the usual requirement of threshold privacy in secret sharing schemes are satisfied by the two GOSS schemes since this property follows essentially from Shamir threshold scheme and Asmuth-Bloom scheme, respectively. Secondly, it is easier for an inside adversary to break the group-oriented property than an outside adversary.

In this paper, we analyze the security of (t,m,n) GOSS schemes of Miao et al. [15,16] and show that these schemes do not satisfy the group oriented property required by GOSS schemes. Note that the original schemes are claimed to be unconditionally secure with share reuse. However, our analysis shows that the schemes are vulnerable against an unauthenticated adversary even in the one-time sense. Precisely, this paper presents the following contributions:

1. We review the polynomial-based (t,m,n) GOSS scheme [15] and show that this scheme does not guarantee the group oriented property. Specifically, we propose an attack which allows an outside adversary with no valid share to obtain the shared secret provided that she observes t or more honest RCs in the secret reconstruction phase. Interestingly, we can model the problem of recovering the secret after seeing $m-1$ RCs as solving a *variant* of the Learning With Errors (LWE) problem. This observation immediately disproves the claimed unconditional security of the (t,m,n) GOSS scheme, since solving the LWE problem is assumed to be computationally hard [23]. To the best of our knowledge, we are the first to model attacking a GOSS scheme as solving an LWE instance.
2. By employing knowledge on the hardness of LWE, we analyze the impact of parameter choice in the polynomial-based GOSS on the effectiveness of our attack. We further prove that under certain conditions, the adversary can successfully obtain the shared secret s with extremely high probability.
3. We review the CRT-based (t,m,n) GOSS scheme [16] and show that this scheme does not guarantee the group-oriented property. Specifically, we propose an attack method which allows an outside adversary with no valid share to obtain the shared secret provided that she gets more than t RCs in the secret reconstruction phase. We also prove that under certain conditions, the adversary can successfully obtain the shared secret with extremely high probability.
4. We implement our attacks to verify their effectiveness. The experiments show that when the adversary observes t honest shares in the reconstruction phase, her probability of successfully obtaining the shared secret s is about 85% for polynomial-based GOSS and 37% for CRT-based GOSS, respectively. Moreover, this probability increases quite quickly with the number of honest RCs the adversary can obtain. For example, when $t=4$, the adversary's success probabilities for both polynomial-based GOSS and CRT-based GOSS increase to about 99% when she can see 8 honest RCs.

2. Preliminaries

We first introduce some notations. All logarithms (denoted as $\log(n)$) throughout this paper are to the base 2. Column vectors are used throughout this paper unless stated otherwise. We say that a function $\mathrm{negl}(n) : \mathbb{N} \to \mathbb{R}$ is negligible in n, if it decreases faster than any inverse polynomial $1/\mathrm{poly}(n)$ in n; formally, there exists an integer N such that for all $n \geqslant N, |\mathrm{negl}(n)| < 1/\mathrm{poly}(n)$. Denote $[n] = \{1, 2, \ldots, n\}$ the set of all positive integers less than or equal to n. We use the expression $x \overset{\$}{\leftarrow} X$ to denote generation of an element x uniformly at random over its support X. If we have a distribution $\mathscr{D}$ which is different from uniform, by a slight abuse of notation, we will denote generation of an element x according to this distribution as $x \overset{\$}{\leftarrow} \mathscr{D}$. In a secret sharing scheme, the secret s belongs to a secret space $\mathscr{S}$ consisting of a finite number of elements. A secret sharing scheme involves $n+1$ parties. There is a trusted dealer and n users (interchangeably, participants and shareholders) among them. We denote the user set to be $\mathscr{U} = \{U_1, \ldots, U_n\}$. For an index set $I_m = \{i_1, \ldots, i_m\} \subset [n], \mathscr{U}_{I_m} = \{U_{i_1}, \ldots, U_{i_m}\}$ is the set of users who are indexed by I_m. Each user U_i holds a share $s_i \in \mathscr{V}_i$ from the dealer, where $\mathscr{V}_i$ is the share space of user U_i. For the set of users $\mathscr{U}_{I_m}$, their joint share space is denoted as $\mathscr{V}_{I_m} = \mathscr{V}_{i_1} \times \ldots \times \mathscr{V}_{i_m}$. We write it as $\mathscr{V}_{I_m} = \prod_{i \in I_m} \mathscr{V}_i$, for short. Consequently, $\mathscr{V}_{[n]}$ is the joint share space of all users. We also write $\mathscr{V}_{[n]}$ as $\mathscr{V}^n$ for simplicity.

2.1. Shamir's Threshold Scheme

In Shamir's (t, n) threshold scheme, the secret space $\mathscr{S}$ and each user's share space $\mathscr{V}_i$ is a finite field $\mathbb{F}_p$. The scheme consists of the following two algorithms.

- **ShareGen**: It takes as input an element of $\mathbb{F}_p$ and returns an element of $\mathbb{F}_p^n$ ($\mathbb{F}_p \to \mathbb{F}_p^n$, for short). To share a secret $s \in \mathbb{F}_p$, the dealer chooses a random polynomial $f(x) = s + a_1 x + \cdots + a_{t-1} x^{t-1}$ of degree at most $t-1$, where the free coefficient of the polynomial is set as $f(0) = s$ and $a_i \overset{\$}{\leftarrow} \mathbb{F}_p$ for $i = 1, \ldots, t-1$. The share for user U_i is an evaluation of the polynomial at her index $s_i = f(i)$.
- **Reconstruct** : $\mathbb{F}_p^m \to \mathbb{F}_p$. Any collection of m ($m \geqslant t$) authentic shares $(s_{i_1}, \ldots, s_{i_m}) \in \mathscr{V}_{I_m}$ can reconstruct the secret s using the Lagrange interpolation.

2.2. Asmuth-Bloom's Threshold Scheme

In Asmuth-Bloom's threshold scheme, the secret space is a modular ring of size p_0 as $\mathscr{S} = \mathbb{Z}_{p_0}$ and each user's share space is a modular ring of size p_i as $\mathscr{V}_i = \mathbb{Z}_{p_i}$. The integers $(p_0, p_1, \ldots, p_n)$ are chosen with the following constraints:

1. $p_1 < \cdots < p_n$.
2. $p_0 \cdot p_{n-t+2} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$.
3. The integers $(p_0, p_1, \ldots, p_n)$ are co-prime.

Asmuth-Bloom's threshold scheme consists of the following two algorithms.

- **ShareGen** : $\mathbb{Z}_{p_0} \to \prod_{i \in [n]} \mathbb{Z}_{p_i}$. To share a secret $s \in \mathbb{Z}_{p_0}$, the dealer randomly chooses an integer α such that $s + \alpha p_0 \in \mathbb{Z}_{p_1 \cdot \ldots \cdot p_t}$. The share for user U_i is calculated as $s_i = (s + \alpha p_0) \bmod p_i$.
- **Recosntruct** : $\prod_{i \in I_m} \mathbb{Z}_{p_i} \to \mathbb{Z}_{p_0}$. Any collection of m ($m \geqslant t$) authentic shares $(s_{i_1}, \ldots, s_{i_m}) \in \mathscr{V}_{I_m}$ can reconstruct the secret s using the CRT by solving the following modular equation system.

$$\begin{cases} x = s_{i_1} \mod p_{i_1} \\ x = s_{i_2} \mod p_{i_2} \\ \cdots \\ x = s_{i_m} \mod p_{i_m} \end{cases} \tag{1}$$

After obtaining the solution of the above modular equation system, the secret can be reconstructed as $s = x \bmod p_0$.

2.3. The LWE problem

The LWE problem proposed by Regev [23] has become a standard hardness assumption in lattice-based cryptography [24]. We briefly introduce it below.

Let n be a positive integer, q be an odd prime, and let $\mathscr{D}$ be a distribution, which is called error distribution, over the modular integer ring $\mathbb{Z}_q$. Denote by $\mathbf{s}$ a fixed secret vector in $\mathbb{Z}_q^n$ selected according to the uniform distribution on its support. Let $\mathscr{L}_{n,q,\mathscr{D}}$ be the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ generated by choosing $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and an error $e \xleftarrow{\$} \mathscr{D}$, then returning

$$(\mathbf{a}, c) = (\mathbf{a}, \mathbf{a}, \mathbf{s} + e) \tag{2}$$

in $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\cdot,\cdot$ is the inner product of two vectors in $\mathbb{Z}_q^n$. The search LWE problem is to find the secret vector $\mathbf{s}$ given m samples generated from the LWE distribution $\mathscr{L}_{n,q,\mathscr{D}}$.

Matrix Representation of LWE. Given m samples $(\mathbf{a}_i, c_i)$, for $i = 1, \ldots, m$, generated from a given LWE distribution $\mathscr{L}_{n,q,\mathscr{D}}$, we can rewrite the samples into a matrix representation. Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ be a matrix whose rows consist of the m samples of $\mathbf{a}_i$, let $\mathbf{c} = (c_1, \ldots, c_m) \in \mathbb{Z}_q^m$, and let an error vector be constructed as $\mathbf{e} = (e_1, \ldots, e_m) \xleftarrow{\$} \mathscr{D}^m$. Thus, the search LWE problem can be viewed as solving the following linear modular equation system with noise:

$$\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e}. \tag{3}$$

Here $\mathbf{s} \in \mathbb{Z}_q^n$ is the unknown vector, and "+" denotes an element-wise vector summation over $\mathbb{Z}_q$.

Typically, in lattice-based cryptography, the errors are sampled following the discrete Gaussian distribution [25]. However, we note that any discrete distribution with small width is fine for our purpose in this paper. Generally speaking, high-dimensional LWE problem is assumed to be intractable [23,26]. Estimations on the concrete hardness of solving LWE instances can be found, e.g., in [27,28]. We highlight some useful observations on the concrete hardness of LWE problem.

- The length of the secret vector $\mathbf{s}, n$, is called the dimension of the LWE problem. The problem becomes harder when the dimension n increases. The LWE problem is only intractable for large enough n.
- The relative error rate of the LWE problem, $\alpha = |e|/q$, reflects the relative magnitude of the errors $e_i \xleftarrow{\$} \mathscr{D}$ compared with the module q. The concrete hardness of LWE problem increases with the growth of the relative error rate.
- The number of equations (or LWE samples) is m. The probability of finding a unique solution to the LWE problem increases with m. At the same time, the computational effort decreases with m.

3. Miao et al.'s GOSS Schemes

An unauthenticated outside adversary can join the reconstruction phase of a secret sharing scheme and obtain the secret from honest shares. In order to deal with this problem, Miao et al. proposed the notion of GOSS scheme. In this section, we first introduce the syntax and properties of (t, m, n) GOSS and then review two concrete constructions.

3.1. Syntax and Properties of (t, m, n) GOSS

A (t, m, n) GOSS scheme consists of three algorithms (**ShareGen**, **RCGen**, **Reconstruct**). The GOSS scheme involves a dealer D and a set of n users $\mathscr{U} = \{U_1, \ldots, U_n\}$. Besides the usual secret space $\mathscr{S}$ and share space $\mathscr{V}_i$ in a threshold scheme, GOSS has another space $\mathscr{RC}$ called the space of *randomized components* (RC, for short). Roughly speaking, RC in GOSS is a share scrambled with some randomized noise such that it is supposed to leak no information about the secret while having the same effect as a share in secret reconstruction.

- **ShareGen** : $\mathscr{S} \to \mathscr{V}^n$. This algorithm splits a secret $s \in \mathscr{S}$ into n shares, and distributes each share $s_i \in \mathscr{V}_i$ to user U_i.
- **RCGen** : $\mathscr{V}_{I_m} \to \mathscr{RC}_{I_m}$. This algorithm takes a set of m shares $(s_{i_1}, \ldots, s_{i_m})$ as input and generates m RCs $(c_{i_1}, \ldots, c_{i_m})$, where $t \leqslant m \leqslant n$ and $I_m \subset [n]$. For $j \in [m]$, the RC c_{i_j} belongs to user $U_{i_j} \in \mathscr{U}_{I_m}$.
- **Reconstruct** : $\mathscr{RC}_{I_m} \to \mathscr{S}$. This algorithm reconstructs the secret s from m RCs $(c_{i_1}, \ldots, c_{i_m})$, where $t \leqslant m \leqslant n$ and $I_m \subset [n]$.

The required properties of (t, m, n) GOSS are informally listed below. The first two of them are the same as those of unconditionally secure threshold schemes.

- Correctness. When $m \geqslant t$, the secret s can be correctly reconstructed from m RCs $(c_{i_1}, \ldots, c_{i_m})$ if they are all correct.
- Perfect secrecy. When $m < t$, the m shares $(s_{i_1}, \ldots, s_{i_m})$ leak no information about the secret s even for adversary with unlimited computation power.
- Group oriented property. This property is an extra requirement which is specific to GOSS. Once m $(m \geqslant t)$ shareholders form a tightly coupled group by generating RCs, the secret s can be recovered from these m RCs if and only if all of them are valid. In other words, if there are invalid RCs among m RCs in the **Reconstruct** algorithm, the output of the algorithm is not the original secret s. This property requires all users joining the reconstruction phase to have valid RCs (thus shares) in order to reconstruct the secret. According to Miao et al. [15], if the secret is reconstructed from RCs instead of shares, the threshold actually becomes m.

Remark 1. We note that in the syntax of GOSS the parameter m is not essential since the only requirement of m is that $m \geqslant t$, hence m may be ignored for brevity. However, we stick to the notion of (t, m, n) GOSS to be consistent with the notation used in the original paper by Miao et al.

3.2. (t, m, n) GOSS Based on Shamir's Threshold Scheme

In this subsection, we briefly review the GOSS scheme proposed by Miao et al. [15] which is based on Shamir's threshold scheme. The GOSS scheme shares the secret s according to that of Shamir's threshold scheme except that the secret s belongs to a smaller field $\mathbb{F}_q$, while the coefficients of the polynomial belong to a larger field $\mathbb{F}_p$. To share a secret $s \in \mathbb{F}_q$, the dealer chooses a random polynomial $f(x) = s + a_1 x + a_{t-1} x^{t-1} \bmod p$, where $a_i \overset{\$}{\leftarrow} \mathbb{F}_p$ for $i = 1, \ldots, t-1$ with $p > q + nq^2$. The share for user U_i is $f(x_i)$. In the reconstruction phase, a set of m users $\mathscr{U}_{I_m} = \{U_{i_1}, \ldots, U_{i_m}\}$ first generate RCs $c_{i_j} = \left(f\left(x_{i_j}\right) \prod_{v=1, v \neq j}^{m} \frac{-x_{i_v}}{x_{i_j} - x_{i_v}} + r_{i_j} q \right) \bmod p$. The first part of an RC is nothing but the reconstruction component in Shamir's threshold scheme. The second part of an RC is a random component $r_{i_j} q$, where $r_{i_j} \overset{\$}{\leftarrow} \mathbb{F}_p$ is a uniformly random element. After computing the RCs, all the m users reconstruct the secret s by summing up their RCs as $s = \sum_{j=1}^{m} c_{i_j} \bmod q$. The detailed steps of the GOSS scheme by Miao et al. are presented in Fig. 1.

3.3. (t, m, n) GOSS Based on Asmuth-Blomm's Threshold Scheme

In this subsection, we briefly review the GOSS scheme proposed by Miao et al. [16] which is based on Asmuth-Blomm's threshold scheme. We call this scheme CRT-based GOSS scheme for short throughout this paper. The CRT-based GOSS scheme shares the secret s according to that of Asmuth-Bloom's threshold scheme. The dealer first picks n integers $p_1 < \ldots < p_n$, such that $p_n^2 \cdot p_{n-t+2} \cdot \cdots \cdot p_n < p_1 \cdot p_2 \cdots p_t, np_0^3/(p_0 - 1) < p_1$ and $\gcd(p_i, p_j) = 1$, $(i, j = 0, \ldots, n$ and $i \neq j)$. The integer p_i is called public modulus associated with each user U_i. To share a secret $s \in \mathbb{Z}_{p_0}$, the dealer chooses an integer $s + \alpha p_0 \in \mathbb{Z}_{\lfloor (p_1 \cdot \ldots \cdot p_t)/p_0 \rfloor}$. Then the dealer computes $s_i = s + \alpha p_0 \bmod p_i$, $i = 1, \ldots, n$ and sends the share s_i to user U_i. At reconstruction phase, a set of m users $\mathscr{U}_{I_m} = \{U_{i_1}, \ldots, U_{i_m}\}$ first generate RCs $c_{i_j} = \left(s_{i_j} \frac{N}{p_{i_j}} y_{i_j} + r_{i_j} \frac{N}{p_{i_j}} p_0 \right) \bmod N$, where $N = \prod_{j=1}^{m} p_{i_j}$ and y_{i_j} is the inverse of $\frac{N}{p_{i_j}}$ in the modular ring $\mathbb{Z}_{p_{i_j}}$. The first part of an RC is the reconstruction component in Asmuth-Bloom's threshold scheme. The second part of an RC is a random component $r_{i_j} \frac{N}{p_{i_j}} p_0$ where $r_{i_j} \overset{\$}{\leftarrow} \mathbb{Z}_{p_0}$. After computing the RCs, all the m users reconstruct the secret s by summing up their RCs as $s = \sum_{j=1}^{m} c_{i_j} \bmod p_0$. The detailed steps of CRT-based GOSS by Miao et al. are presented in Fig. 2.

4. Cryptanalysis of Polynomial-based GOSS Scheme

In this section, we present a detailed cryptanalysis of the polynomial-based GOSS scheme [15]. The polynomial-based GOSS scheme is claimed to be unconditionally secure with respect to the group oriented property.

Claim 1 Theorem 2 of Miao et al. [15]. Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. Then there exists an integer q_0 such that for any $q > q_0$ with $\mathscr{S} = \mathbb{F}_q$, we have, for any positive number ϵ,

$$H(s) - H(s|C_{I_k}) \leqslant \epsilon, \tag{4}$$

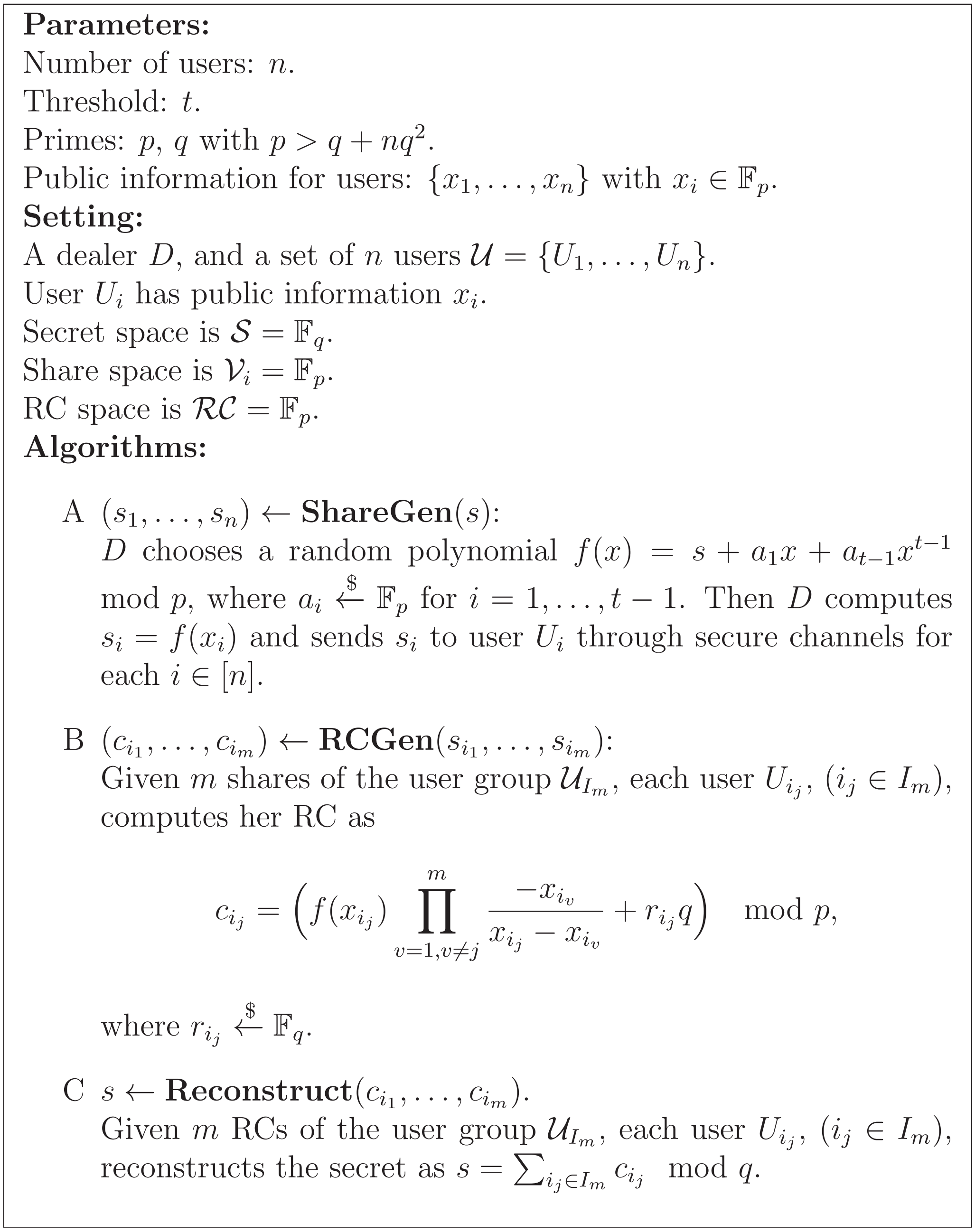
where $H(s)$ is the Shannon entropy of the secret s and $H(s|C_{I_k})$ is the conditional entropy of s given the k RCs C_{I_k}.

However, we show that the scheme is not unconditionally secure. Specifically we develop attacks against the GOSS scheme. Our attacks allow an unauthenticated adversary to reconstruct the secret with high probability provided that the adversary sees at least t authentic RCs in the reconstruction phase.

4.1. Model the Attack as Solving an LWE Problem

Consider an adversary $\mathscr{A}$ who sees $k \geqslant t$ honest RCs during the reconstruction phase in the polynomial-based GOSS. In this subsection we show how to model the adversary's task of obtaining the shared secret s to solving a variant of LWE instance.

Assume that there are m users $U_{I_m} = (U_{i_1}, \ldots, U_{i_m})$ at the reconstruction phase. Now we present the attack strategy for an outside adversary $\mathscr{A}$ who knows k $(k \geqslant t)$ RCs $C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$ with $I_k \subset I_m$. Denote the k authentic users, whose RCs are observed by the adversary $\mathscr{A}$, as $U_{I_k} = (U_{i_1}, \ldots, U_{i_k})$. Note that the adversary knows public information of all the m users as $X_{I_m} = (x_{i_1}, \ldots, x_{i_m})$. Denote the shares of U_{I_k} as $S_{I_k} = (s_{i_1}, \ldots, s_{i_k})$.



Parameters:

Number of users: n.

Threshold: t.

Primes: p, q with $p > q + nq^2$.

Public information for users: $\{x_1, \ldots, x_n\}$ with $x_i \in \mathbb{F}_p$.

Setting:

A dealer D, and a set of n users $\mathcal{U} = \{U_1, \ldots, U_n\}$.

User U_i has public information x_i.

Secret space is $\mathcal{S} = \mathbb{F}_q$.

Share space is $\mathcal{V}_i = \mathbb{F}_p$.

RC space is $\mathcal{RC} = \mathbb{F}_p$.

Algorithms:

A $(s_1, \ldots, s_n) \leftarrow$ **ShareGen**(s):
D chooses a random polynomial $f(x) = s + a_1 x + a_{t-1} x^{t-1} \mod p$, where $a_i \xleftarrow{\$} \mathbb{F}_p$ for $i = 1, \ldots, t-1$. Then D computes $s_i = f(x_i)$ and sends s_i to user U_i through secure channels for each $i \in [n]$.

B $(c_{i_1}, \ldots, c_{i_m}) \leftarrow$ **RCGen**$(s_{i_1}, \ldots, s_{i_m})$:
Given m shares of the user group $\mathcal{U}_{I_m}$, each user U_{i_j}, $(i_j \in I_m)$, computes her RC as

$$c_{i_j} = \left(f(x_{i_j}) \prod_{v=1, v \neq j}^{m} \frac{-x_{i_v}}{x_{i_j} - x_{i_v}} + r_{i_j} q\right) \mod p,$$

where $r_{i_j} \xleftarrow{\$} \mathbb{F}_q$.

C $s \leftarrow$ **Reconstruct**$(c_{i_1}, \ldots, c_{i_m})$.
Given m RCs of the user group $\mathcal{U}_{I_m}$, each user U_{i_j}, $(i_j \in I_m)$, reconstructs the secret as $s = \sum_{i_j \in I_m} c_{i_j} \mod q$.

Fig. 1. Detailed construction of polynomial-based (t, m, n) GOSS scheme.

Recall that the share for U_{i_j} where $i_j \in I_k$ is $s_{i_j} = f\left(x_{i_j}\right)$ and $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$. Set $\mathbf{s} = (s, a_1, \ldots, a_{t-1})$ and $\mathbf{x}_{i_j} = \left(1, x_{i_j}, \ldots, x_{i_j}^{t-1}\right)$. Then we have $s_{i_j} = f\left(x_{i_j}\right) = \mathbf{x}_{i_j}, \mathbf{s}$. When m shareholders collaborate to reconstruct the secret, the randomized components are computed as

$$c_{i_j} = \left(f\left(x_{i_j}\right) \prod_{v=1, v \neq j}^{m} \frac{-x_{i_v}}{x_{i_j} - x_{i_v}} + r_{i_j} q\right) \bmod p,$$

for each $i_j \in I_m$. We further use the notation $l_{m,j} = \prod_{v=1, v \neq j}^{m} \frac{-x_{i_v}}{x_{i_j} - x_{i_v}}$ to simplify the expression and call it Lagrange coefficient since it is related to Lagrange interpolation. Now we have

Parameters:

Number of users: n.

Threshold: t.

Primes: $p_0, \ldots, p_n$ with $p_0^2 p_{n-t+2} \cdots p_n < p_1 p_2 \cdots p_t$, $np_0^3/(p_0-1) < p_1$ and $\gcd(p_i, p_j) = 1$, $(i, j = 0, \ldots, n$ and $i \neq j)$.

Public information for users: $\{x_1, \ldots, x_n\}$,
with $x_i \in \mathbb{F}_{p_i}$.

Setting:

A dealer D, and a set of n users $\mathcal{U} = \{U_1, \ldots, U_n\}$.

User U_i has public information x_i.

A user group $\mathcal{U}_{I_m}$ of size m try to reconstruct the secret.

Secret space is $\mathcal{S} = \mathbb{Z}_{p_0}$.

Share space is $\mathcal{V}_i = \mathbb{Z}_{p_i}$.

RC space is $\mathcal{RC} = \mathbb{Z}_N$, where $N = \prod_{i_j \in I_m} p_{i_j}$.

Algorithms:

A $(s_1, \ldots, s_n) \leftarrow \textbf{ShareGen}(s)$:
D chooses an integer α such that $s + \alpha p_0 \in \mathbb{Z}_{\lceil (p_1 \cdot \ldots \cdot p_t)/p_0 \rceil}$. Then D computes $s_i = s + \alpha p_0 \mod p_i$ and sends s_i to user U_i through secure channels for each $i \in [n]$.

B $(c_{i_1}, \ldots, c_{i_m}) \leftarrow \textbf{RCGen}(s_{i_1}, \ldots, s_{i_m})$:
Given m shares of the user group $\mathcal{U}_{I_m}$, each user U_{i_j}, $(i_j \in I_m)$, computes her RC as

$$c_{i_j} = (s_{i_j} \frac{N}{p_{i_j}} y_{i_j} + r_{i_j} \frac{N}{p_{i_j}} p_0) \mod N,$$

where $N = \prod_{i_j \in I_m} p_{i_j}$ and $r_{i_j} \xleftarrow{\$} \mathbb{Z}_{p_0}$.

C $s \leftarrow \textbf{Reconstruct}(c_{i_1}, \ldots, c_{i_m})$.
Given m RCs of the user group $\mathcal{U}_{I_m}$, each user U_{i_j}, $(i_j \in I_m)$, reconstructs the secret as $s = \sum_{i_j \in I_m} c_{i_j} \mod p_0$.

Fig. 2. Detailed construction of CRT-based (t, m, n) GOSS scheme.

$$c_{i_j} = l_{m,j}\mathbf{x}_{i_j}, \mathbf{s} + r_{i_j} q \bmod p, \tag{5}$$

for $i_j \in I_k$. Recall that q and p are primes. Hence $\gcd(q, p) = 1$ and we can multiply both sides of Eq. (5) by $q^{-1} \bmod p$ to get

$$q^{-1} c_{i_j} = q^{-1} l_{m,j}\mathbf{x}_{i_j}, \mathbf{s} + r_{i_j} \bmod p. \tag{6}$$

Now we rewrite Eq. (6) as

$$\left(q^{-1} l_{m,j}\mathbf{x}_{i_j}, q^{-1} c_{i_j}\right) = \left(q^{-1} l_{m,j}\mathbf{x}_{i_j}, q^{-1} l_{m,j}\mathbf{x}_{i_j}, \mathbf{s} + r_{i_j}\right) \bmod p, \tag{7}$$

for $i_j \in I_k$. Note that $r_{i_j} \in \mathbb{F}_q$ and $q < p$. If we think of r_{i_j} as a random error taken from the uniform distribution over $\mathbb{F}_q$, $\mathbf{s}$ as the secret vector of size t and $q^{-1} l_{m,j} \mathbf{x}_{i_j}$ as a known random vector of size t, then $\left(q^{-1} l_{m,j} \mathbf{x}_{i_j}, q^{-1} c_{i_j}\right)$ can be viewed as being sampled from a variant of LWE distribution $\mathscr{L}_{t,q,\mathscr{D}}$ (please refer to Eq. (2) for a close comparison). Write $\mathbf{X} = \left[q^{-1} l_{m,1} \mathbf{x}_{i_1}^T, \ldots, q^{-1} l_{m,k} \mathbf{x}_{i_k}^T\right], \mathbf{c} = \left[q^{-1} c_{i_1}, \ldots, q^{-1} c_{i_k}\right]$ and $\mathbf{r} = \left[r_{i_1}, \ldots, r_{i_k}\right]$, then we get a matrix representation of the LWE problem

$$\mathbf{c} = \mathbf{X}\mathbf{s} + \mathbf{r}, \tag{8}$$

where $\mathbf{X} \in \mathbb{F}_p^{k \times t}, \mathbf{s} \in \mathbb{F}_q \times \mathbb{F}_p^{t-1}, \mathbf{r} \in \mathbb{F}_q^k$ and $\mathbf{c} \in \mathbb{F}_p^k$ (compare Eq. (8) with Eq. (3)). Notice that the vector $\mathbf{r}$ is uniformly sampled from $\mathbb{F}_q^k$, which is much shorter than a random vector in $\mathbb{F}_p^k$ on average (because $p > q + nq^2$).

Remark 2. Note that although $\left(q^{-1} l_{m,j} \mathbf{x}_{i_j}, q^{-1} c_{i_j}\right)$ takes the same form as an LWE sample, it does not exactly follow the standard LWE distribution as defined in Section 2.3 for two reasons. Firstly, standard LWE distribution uses discrete Gaussian distribution as the error distribution $\mathscr{D}$, but from Eq. (7) we can see that r_{i_j} is uniformly distributed over $\mathbb{F}_q$. Nonetheless, since $p > n^2 q + q$, r_{i_j} can be safely regarded as a small error. Secondly, standard LWE distribution requires the vector $\mathbf{a} \in \mathbb{Z}_p^n$ to be uniform while in Eq. (7) the vector $q^{-1} l_{m,j} \mathbf{x}_{i_j}$ is not uniformly random as $\mathbf{x}_{i_j} = \left(1, x_{i_j}, \ldots, x_{i_j}^{t-1}\right)$ is highly structured.

However, the difference between standard LWE and the variation of LWE described by Eq. (7) is not significant for our purpose. The requirement of uniform distribution of $\mathbf{a}$ and discrete Gaussian distribution (in practice) of the error e in standard LWE problem is to guarantee computational hardness of LWE so that it can be used to construct computationally secure cryptographic primitives. But in this work we model the problem of an outside adversary against the group oriented property of GOSS scheme as solving a variant of LWE problem. Thus the adversary can actually solve the problem given enough computational power. Precisely, if the LWE problem $\mathscr{L}_{t,q,\mathscr{D}}$ described by Eq. (8) has a unique solution, then the adversary can get the correct secret s shared by the dealer. Note that we are not trying to prove that security of the polynomial-based GOSS can be reduced to the hardness of our variant of LWE problem. On the contrary, the message we want to convey is that: if the adversary can find a unique solution of the LWE instance, then she can recover the shared secret. In fact, we do not know how to prove that our LWE variant is easy to solve. However, we show that the adversary's effort to break the group-oriented property is upper-bounded by the hardness of the variant LWE problem.

4.2. Discussions on Parameter Choice and Its Impact

The observation that the attack can be modeled as solving an LWE problem in the last subsection shows, *straightforwardly*, that the (t, m, n) GOSS scheme is not unconditionally secure. In this subsection, we first discuss how the choice of parameters in the GOSS scheme affects the practical effectiveness of our attack. Then we formally disprove Claim 1.

Since we have modeled the adversary's task as solving Eq. (8), we discuss how the choice of parameters in the (t, m, n) GOSS scheme affect the concrete hardness of the LWE problem. From Eq. (8), we can see that the dimension of the LWE problem is t and the relative error rate of the LWE problem α has a positive correlation with q/p. In addition, k is the number of LWE samples. Recalling the observations listed at the end of Section II, we conclude that the concrete hardness of the LWE problem increases as t and $\alpha \underset{\sim}{\propto} q/p$ increases[1]. Note that t is the threshold of the GOSS scheme and, according to Fig. 1, q is the size of secret space and $p > q(nq + 1)$ is the share space size. The number of LWE samples k equals the number of honest RCs the adversary can obtain at the reconstruction phase.

Regarding the impact of parameter choice in the (t, m, n) GOSS scheme on effectiveness of our attack, we have the following observations.

- The (t, m, n) GOSS scheme is practically insecure if the threshold t is small. This is because LWE problem is easy to solve when the dimension is small. To be specific, modern cryptographic protocols [29] using LWE problem as hardness assumption usually require the LWE dimension to be at least 1024.
- The relative error rate α is approximately proportional to q/p and $p > q(nq + 1)$. For fixed number of users n, larger secret size q results in a smaller α. The same correlation occurs between α and n, when q is fixed. In other words, the concrete hardness of the LWE problem decreases as n or q increases. In the context of the (t, m, n) GOSS scheme, when the threshold t is fixed, it is easier for the adversary to succeed if the number of total users n is larger, or if the secret size q is larger.
- The greater the number of samples k, the easier is it for the adversary to solve the LWE problem in the following two senses: 1) the probability of getting a unique solution increases; and 2) the computational effort reduces. In the context of the (t, m, n) GOSS scheme, when the threshold t, the number of total users n and the secret size q are fixed, the adversary's success probability increases with the number of honest RCs she can obtain. In addition, obtaining more RCs can reduce her computational cost.

[1] Here the symbol means approximately proportional to.

Next we formally disprove Claim 1. Following Quisquater et al. [30], we use the notion of *loss of entropy of the secret* as a measure of the adversary's advantage against the group-oriented property. Loosely speaking, the loss of entropy of the secret generated by the knowledge of a set of RCs indicates the information the adversary gets about the original secret s. When the adversary can get no extra information about the secret, the loss of entropy of the secret is zero. While in the extreme case when the adversary can uniquely determine the shared secret, the loss of entropy of the secret achieves its maximal value which is the entropy of the secret. We present formal definition of the loss of entropy of the secret in Definition 1 adapted from Definition 2 of Quisquater et al. [30].

Definition 1. Let $(\mathbf{ShareGen}(), \mathbf{RCGen}(), \mathbf{Reconstruct}())$ be a (t, m, n) GOSS. Let $(s_1, \ldots, s_n) \leftarrow \mathbf{ShareGen}(s)$. For $I_m = \{i_1, \ldots, i_m\} \subseteq [n]$, let $(c_{i_1}, \ldots, c_{i_m}) \leftarrow \mathbf{RCGen}(s_{i_1}, \ldots, s_{i_m})$. Set $C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$, where $I_k \subseteq I_m$ and $|I_k| = k \geqslant t$. We define the loss of entropy of the secret s by the knowledge of the RCs C_{I_k} as $\Delta(C_{I_k}) = H(s) - H(s|C_{I_k})$, where $H()$ is the Shannon entropy of the secret s and $H(s|C_{I_k})$ is the conditional entropy of s given the k RCs C_{I_k}.

Remark 3. Our definition differs from that of Quisquater et al. [30] in the sense that they define the loss of entropy of the secret with respect to the knowledge of shares obtained by inside adversary, while we define the loss of entropy of the secret with respect to the knowledge of RCs obtained by outside adversary. This is because we consider an outside adversary who can obtain RCs rather than shares.

Remark 4. The loss of entropy can be negative. In this paper, for the sake of simplicity as well as to be consistent with the original GOSS description, we assume uniform distribution of the secret. In the case of a uniformly distributed secret, the loss of entropy of the secret is always non-negative. However, both our attack strategy and analysis extend, in a straightforward manner, to the case of a general distribution of the secret.

Our main result is formulated as Theorem 1.

Theorem 1. *Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. For any q, there exists a k_0 such that, if $k \geqslant k_0$, then with high probability we have $\Delta(C_{I_k}) = H(s)$.*

Heuristic 1. Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. For any $k \geqslant t$, we have $\Delta(C_{I_k}) > \mathrm{negl}(q)$.

Theorem 1 and Heuristic 1 actually say that there exists an unauthenticated adversary $\mathscr{A}$ who can join the reconstruction phase of the GOSS scheme and successfully obtain the secret s when more than t honest shareholders participant in the reconstruction phase. That is, the GOSS scheme provides basically no better security against unauthenticated outside adversary compared to the plain Shamir threshold scheme.

Since we have modeled the adversary's task of recovering the secret as solving an instance of our LWE variant, our proof of Theorem 1 essentially deals with the characterization of the solutions of the LWE problem. In short, we prove that when the adversary sees enough RCs (which corresponds to enough samples of the LWE instance), the chance that the LWE instance has a unique solution is overwhelmingly large so that she can uniquely recover the shared secret. On the other hand, when the adversary can only obtain limited RCs, the LWE instance might have more than one solutions. However, in most cases, those solutions do not exhibit a uniform distribution. This non-uniform distribution of solutions leaks information on the secret to the adversary. For example, some possible choices for the secret might not be solutions to the LWE instance. Then the adversary can exclude such values as the secret, which means the entropy of the secret is reduced due to the adversary's knowledge of RCs. Specifically, we prove that even in the case that the number of RCs the adversary can get is the threshold t, the loss of entropy of the secret is non-negligible.

We use Proposition 1 in our proof of Theorem 1 and defer the proof of Proposition 1 for later.

Proposition 1. Suppose $p > q + nq^2$, where p, q are primes and let $t < m \leqslant n$. Let $k_0 = \frac{t \log p}{\log\left(\frac{1+nq}{10}\right)}$. Suppose that $\frac{m}{t} \geqslant k \geqslant k_0$. Let $\mathbf{X}$ be the matrix as in Eq. (7). Then the secret $\mathbf{s} \in \mathbb{F}_p^t$ is uniquely determined with high probability by the LWE sample $(\mathbf{X}, \mathbf{c})$, where $\mathbf{c} = \mathbf{X}\mathbf{s} + \mathbf{r} \bmod p$, where $\mathbf{r} \leftarrow \{0, 1, \ldots, q-1\}^k$.

Proof of Theorem 1 Due to Proposition 1, there exists an integer k_0 when $k \geqslant k_0$ the LWE variant has a unique solution with high probability for suitable parameters. When the adversary solves the LWE problem and gets the solution $\mathbf{s}'$, the solution vector $\mathbf{s}'$ must be the vector $\mathbf{s}$ which the dealer used for sharing the secret. Hence the adversary can easily get the original secret s by reading the first element of $\mathbf{s}'$. Thus, we get $H(s|C_{I_k}) = 0)$ when $k \geqslant k_0$ and hence $\Delta(C_{I_k}) = H(s)$.

Justification of Heuristic 1. Heuristic 1 says that when $k \geqslant t, H(s) - H(s|C_{I_k}) > neg(q)$. Without loss of generality we only argue this property for the case when $k = t$. We rewrite the matrix representation of Eq. (8) as follows for convenience:

$$\mathbf{c} = \mathbf{X}\mathbf{s} + \mathbf{r}, \tag{9}$$

where $\mathbf{X} \in \mathbb{F}_p^{t\times t}, \mathbf{s} \in \mathbb{F}_q \times \mathbb{F}_p^{t-1}, \mathbf{r} \in \mathbb{F}_q^t$ and $\mathbf{c} \in \mathbb{F}_p^t$. The adversary $\mathscr{A}$ can now enumerate all possible choices of $\mathbf{r} \in \mathbb{F}_q^t$ and solve Eq. (9). For each possible choice of $\mathbf{r}$, Eq. (9) has a unique solution $\mathbf{s}'$ since matrix $\mathbf{X}$ has rank t. In fact, we can write down the solution as $\mathbf{s}' = \mathbf{X}^{-1}(\mathbf{c} - \mathbf{r})$. Denote the first element of $\mathbf{s}'$ by s'. We now pay attention to s', since it is the supposed candidate secret of the dealer. Set $\mathbf{x}$ as the first row of $\mathbf{X}^{-1}$. It follows that $s' = \mathbf{x}(\mathbf{c} - \mathbf{r})$. In general, for a random $\mathbf{r} \in \mathbb{F}_q^t, s' = \mathbf{x}(\mathbf{c} - \mathbf{r})$ will not be in $\mathbb{F}_q$. Denote by S the multiset of all s' which belong to $\mathbb{F}_q$, i.e., $S = \left\{s' | \exists \mathbf{r} \in \mathbb{F}_q^t \text{s.t.} s' = \mathbf{x}(\mathbf{c} - \mathbf{r}) \in \mathbb{F}_q\right\}$. If each element of $\mathbb{F}_q$ occurs with equal frequency in the multiset S, the distribution of candidate secret s' is uniformly distributed over $\mathbb{F}_q$. Then the adversary cannot get any information about the real secret s according to her observation of t RCs. However, it is usually not the case. For most cases of the choice of $\mathbf{x}$ and $\mathbf{c}$, different elements of $\mathbb{F}_q$ occurs with different frequency in S. The statistical distance between the frequency of S and uniform distribution over $\mathbb{F}_q$ depends on $\mathbf{x}$ and $\mathbf{c}$. We find it difficult to bound this statistical distance. So we present the statement that $\Delta(C_{I_k}) > \text{negl}(q)$ as a heuristic. In our experimental evaluation (Section 6.2) below we show that with proper parameter setting, the adversary $\mathscr{A}$ can obtain the original secret s with probability higher than 85% in the case that $k = t$.

Remark 5. Note that in the proof we only concern the characterization of the solutions of the LWE problem. The problem of how to solve the LWE problem is not an issue for an computation-unbounded adversary as we considered in this paper. Of course, more efficient method for solving the LWE problem results in more efficient adversary. However, investigating whether efficient algorithm exists for such LWE variant is out of scope of this paper.

Now we complete with the proof of Proposition 1. In order to prove Proposition 1, we first introduce some background knowledge about lattices. A lattice in $\mathbb{R}^k$ is a discrete additive subgroup generated by a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$. Equivalently, the lattice $\Lambda(B)$ generated by $\mathbf{B}$ is given by $\Lambda(\mathbf{B}) = \left\{\mathbf{v} | \mathbf{v} = \sum_{i=1}^k z_i \mathbf{b}_i\right\}$, where z_i's are integers. We define $\lambda(\Lambda)$ as the length of the shortest nonzero vector of the lattice $\Lambda(B)$. We are interested in the so called p-ary[2] lattices which are lattices satisfying $p\mathbb{F}_p^m \subset \Lambda \subset \mathbb{Z}^m$. Specifically for any matrix $\mathbf{X} \in \mathbb{Z}^{k\times t}$, define the following lattice:

$$\Lambda_p(\mathbf{X}) = \left\{v \in \mathbb{Z}^k : \mathbf{v} = \mathbf{X}\mathbf{s} \bmod p \text{ for some } s \in \mathbb{F}_p^n\right\}. \tag{10}$$

One way to solve the LWE problem is to reduce it to the problem of finding a closest vector in the lattice. Take the LWE problem defined by Eq. (9) as an example. We have $\mathbf{c} = \mathbf{X}\mathbf{s} + \mathbf{r}$, where $\mathbf{r}$ can be viewed as the error vector following the uniform distribution over $\mathbb{F}_q^k$. We observe that the length of $\mathbf{r}$ is relatively small since $q \ll p$. Consider the p-ary lattice $\Lambda_p(\mathbf{X})$ defined in Eq. (10). Then the vector $\mathbf{c}$ is bounded in distance from a vector $\mathbf{v} \in \Lambda_p(\mathbf{X})$. Finding the vector $\mathbf{v}$ from the lattice $\Lambda_p(\mathbf{X})$ given $\mathbf{c}$ is called the closest vector problem. And once $\mathbf{v}$ is found, the secret vector $\mathbf{s}$ can be solved using Gaussian elimination.

We introduce the following lemmas before proving Proposition 1.

Lemma 1. *Let $k > t$, and*

$$\mathbf{X} = \begin{bmatrix} q^{-1}\ell_{m,1} & q^{-1}\ell_{m,1}x_{i_1} & \cdots & q^{-1}\ell_{m,1}x_{i_1}^{t-1} \\ q^{-1}\ell_{m,2} & q^{-1}\ell_{m,2}x_{i_2} & \cdots & q^{-1}\ell_{m,2}x_{i_2}^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ q^{-1}\ell_{m,k} & q^{-1}\ell_{m,k}x_{i_k} & \cdots & q^{-1}\ell_{m,k}x_{i_k}^{t-1} \end{bmatrix} \in \mathbb{F}_p^{k\times t}.$$

Denote $\Lambda_p(\mathbf{X})$ be the p-ary lattice generated by the columns of $\mathbf{X}$, that is,

$$\Lambda_p(\mathbf{X}) = \left\{\mathbf{y} \in \mathbb{Z}^k : \mathbf{y} = \mathbf{X}\mathbf{s} \bmod p \text{ for some } \mathbf{s} \in \mathbb{F}_p^t\right\}.$$

Then $\det(\Lambda_p(\mathbf{X})) = p^{k-t}$.

Proof of Lemma 1 Notice that we can write $\mathbf{X} = q^{-1}\mathbf{L}\mathbf{V}$, where

$$\mathbf{L} = \begin{bmatrix} \ell_{m,1} & 0 & \cdots & 0 \\ 0 & \ell_{m,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \ell_{m,k} \end{bmatrix}$$

and

[2] In the cryptographic literature, the terminology is q-ary lattice. We use p-ary lattice here because in the GOSS schemes, the modulo used by Miao et al. is p.

$$\mathbf{V} = \begin{bmatrix} 1 & x_{i_1} & \cdots & x_{i_1}^{t-1} \\ 1 & x_{i_2} & \cdots & x_{i_2}^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i_k} & \cdots & x_{i_k}^{t-1} \end{bmatrix}.$$

Now, we write

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 \end{bmatrix},$$

where

$$\mathbf{L}_1 = \begin{bmatrix} \ell_{m,1} & 0 & \cdots & 0 \\ 0 & \ell_{m,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \ell_{m,t} \end{bmatrix}$$

and

$$\mathbf{L}_2 = \begin{bmatrix} \ell_{m,t+1} & 0 & \cdots & 0 \\ 0 & \ell_{m,t+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \ell_{m,k} \end{bmatrix}.$$

Also, we write

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix},$$

where

$$\mathbf{V}_1 = \begin{bmatrix} 1 & x_{i_1} & \cdots & x_{i_1}^{t-1} \\ 1 & x_{i_2} & \cdots & x_{i_2}^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i_t} & \cdots & x_{i_t}^{t-1} \end{bmatrix}$$

and

$$\mathbf{V}_2 = \begin{bmatrix} 1 & x_{i_{t+1}} & \cdots & x_{i_{t+1}}^{t-1} \\ 1 & x_{i_{t+2}} & \cdots & x_{i_{t+2}}^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i_k} & \cdots & x_{i_k}^{t-1} \end{bmatrix}.$$

Thus, both $\mathbf{L}_1$ and $\mathbf{V}_1$ are invertible $t \times t$matrices over $\mathbb{F}_p$ since the x_i are assumed to be distinct. Suppose $\mathbf{y} \in \Lambda_p(\mathbf{X})$ and write $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$, where $\mathbf{y}_1 \in \mathbb{Z}^t$ and $\mathbf{y}_2 \in \mathbb{Z}^{k-t}$. Therefore there exists $\mathbf{s} \in \mathbb{F}_p^t$ such that $\mathbf{y} = \mathbf{X}\mathbf{s}\text{mod}p$, that is, $\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = q^{-1}\begin{bmatrix} \mathbf{L}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 \end{bmatrix}\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}\mathbf{s}$ mod p. Hence, we have

$$\mathbf{y}_i = q^{-1}\mathbf{L}_i\mathbf{V}_i\mathbf{s}\text{mod}p$$

for $i = 1, 2$. So, we may write $\mathbf{s} = q\mathbf{V}_1^{-1}\mathbf{L}_1^{-1}\mathbf{y}_1$ mod p and thus, we get $\mathbf{y}_2 = \mathbf{L}_2\mathbf{V}_2\mathbf{V}_1^{-1}\mathbf{L}_1^{-1}\mathbf{y}_1$ mod p. Suppose $\mathbb{z} \in \mathbb{Z}^{k-t}$ such that

$$\mathbf{y}_2 = \mathbf{L}_2\mathbf{V}_2\mathbf{V}_1^{-1}\mathbf{L}_1^{-1}\mathbf{y}_1 + p\mathbb{z}.$$

Thus, we get

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{L}_2\mathbf{V}_2\mathbf{V}_1^{-1}\mathbf{L}_1^{-1} & p\mathbf{I}_{k-t} \end{bmatrix}\begin{bmatrix} \mathbf{y}_1 \\ \mathbb{z} \end{bmatrix},$$

where $\mathbf{I}_t$ is the $t \times t$ identity matrix. Hence, $\Lambda_p(\mathbf{X})$ has basis consisting of the columns of the matrix

$$\mathbf{B} := \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{L}_2\mathbf{V}_2\mathbf{V}_1^{-1}\mathbf{L}_1^{-1} & p\mathbf{I}_{k-t} \end{bmatrix}$$

and

$$\det(\Lambda_p(\mathbf{X})) = |\det(B)| = p^{k-t}$$

Lemma 2. *Suppose $t < k$ are positive integers and p be a prime such that $p^{1-\frac{t}{k}} \geqslant 45$. Then for any uniformly random chosen matrix $\mathbf{A} \in \mathbb{F}_p^{k\times t}$ with $\det(\Lambda_p(\mathbf{A})) = p^{k-t}$, we have*

$$\Pr\left[\lambda_1(\Lambda_p(\mathbf{A})) \leqslant \frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}\right] < 2^{-k},$$

where $\lambda_1(\Lambda_p(\mathbf{A}))$ is the length of the shortest nonzero vector in the p-ary lattice $\Lambda_p(\mathbf{A})$.

Proof of Lemma 2. Let $R = \frac{1}{9}\sqrt{k}p^{1-\frac{t}{k}}$ and consider the k-dimensional ball $B_R(\mathbf{0})$. Notice that the inequality $p^{1-\frac{t}{k}} \geqslant 45$ is equivalent to $R \geqslant \frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}} + \frac{\sqrt{k}}{2}$. Since cubes of unit length centered at integer points tile the entire space $\mathbb{R}^n$, adding $\frac{\sqrt{k}}{2}$, which is the half of the length of the diagonal of the cube of unit length, to $\frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}$ ensures that all integer points with norm at most $\frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}$ lie inside $B_R(\mathbf{0})$. Thus, the number of points in $\mathbb{Z}^k$ with norm at most $\frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}$ is bounded above by the volume of $B_R(\mathbf{0})$. Also, note that the probability that a random integer point is a point in $\Lambda_p(\mathbf{A})$ is $\frac{1}{p^{k-t}} = p^{t-k}$. Hence, by the union bound and Stirling approximation to the Gamma function, we get

$$\begin{aligned} \Pr\left[\lambda_1(\Lambda_p(\mathbf{A})) \leqslant \tfrac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}\right] &\leqslant \mathbf{vol}(B_R(\mathbf{0}))\cdot p^{t-k} \\ &= \tfrac{\pi^{k/2}R^k}{\Gamma(1+k/2)}\cdot p^{t-k} \\ &\leqslant \left(\tfrac{\sqrt{2\pi e}}{9}\right)^k \\ &< 2^{-k}. \end{aligned}$$

Proof of Proposition 1 Suppose $(\mathbf{s}_i, \mathbf{r}_i), i = 1, 2$ are secret-error tuples where $\mathbf{s}_i \in \mathbb{F}_p^t, \mathbf{r}_i \in \mathbb{F}_q^k$ and $\mathbf{s}_1 \neq \mathbf{s}_2$ such that

$$\mathbf{X}\mathbf{s}_1 + \mathbf{r}_1 = \mathbf{c} = \mathbf{X}\mathbf{s}_2 + \mathbf{r}_2 \bmod p.$$

Thus, $\|\mathbf{X}(\mathbf{s}_1 - \mathbf{s}_2)\| = \|\mathbf{r}_2 - \mathbf{r}_1\| \leqslant (q-1)\sqrt{k}$. Since $\mathbf{s}_1 \neq \mathbf{s}_2$, then $\mathbf{X}(\mathbf{s}_1 - \mathbf{s}_2)$ is a nonzero lattice point in $\Lambda_p(\mathbf{X})$ and thus we have that $\lambda_1(\Lambda_p(\mathbf{X})) \leqslant (q-1)\sqrt{k}$. Since $p > q(1+nq)$, then $q - 1 < \frac{p}{1+nq} = \frac{1}{10}p^{1-\frac{t}{k_0}}$. Since $k \geqslant k_0$ then we get $\lambda_1(\Lambda_p(\mathbf{X})) \leqslant (q-1)\sqrt{k} \leqslant \frac{1}{10}\sqrt{k}p^{1-\frac{t}{k_0}} \leqslant \frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}$. Notice that $\mathbf{X}$ is determined by m values $x_{i_1}, \ldots, x_{i_m}$ and $\mathbf{X}$ has kt entries. Since $m \geqslant kt$, that is, the number of values that determines $\mathbf{X}$ exceeds the degrees of freedom of any matrix in $\mathbb{F}_p^{k\times t}$, then we may view $\mathbf{X}$ as a uniformly random matrix in $\mathbb{F}_p^{k\times t}$ and by Lemma 2, the probability that $\lambda_1(\Lambda_p(\mathbf{X})) \leqslant \frac{1}{10}\sqrt{k}p^{1-\frac{t}{k}}$ is less than 2^{-k}. Hence, the secret $\mathbf{s}$ is unique with probability greater than $1 - 2^{-k}$.

Remark 6. We leave some remarks regarding the proof of Proposition 1.

- In the proof of Lemma 2 and hence Proposition 1, we require that $m > kt$. The reason for such requirement is to circumvent the issue that the matrix $\mathbf{X}$ is not drawn uniformly from its support. Otherwise, we need to face the open problem of estimating the length of the shortest non-zero vector in a lattice. Note that the physical meaning of m is the number of users in the reconstruction phase, and k counts how many honest RCs the adversary obtains. The threshold of the GOSS scheme is t. Looking at the proposition, one might think that in order to get a unique solution the adversary needs to gather a large number of users to participate in the reconstruction phase due to the requirement of $m > kt$. Although the proof of Proposition 1 does not cover the case $m \leqslant kt$ due to technical subtlety, we show in the next section that even in this case, our attack is successful for some practical parameters.
- In the proof of Proposition 1, we proved that for certain parameter choice, there exists a unique solution $\mathbf{s} \in \mathbb{F}_p^t$ to Eq. (9). But, actually the solution of Eq. (9) belongs to $\mathbb{F}_q \times \mathbb{F}_p^{t-1}$. This observation leads to the following argument. Even for some parameter choice, Eq. (9) has more than one solutions over $\mathbf{s} \in \mathbb{F}_p^t$, which may not be in $\mathbb{F}_q \times \mathbb{F}_p^{t-1}$. Therefore, it is possible that there is a unique solution over $\mathbb{F}_q \times \mathbb{F}_p^{t-1}$. In that case, the adversary can still successfully get the original secret shared by the dealer.

5. Cryptanalysis of CRT-Based GOSS Scheme

The CRT-based (t, m, n) GOSS scheme proposed by Miao et al. [16] has the group-oriented property as presented in Claim 2.

Claim 2 (*Theorem 2 of Miao et al. [16]*). Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the CRT-based (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. Then for any positive number ϵ, we have

$$H(s) - H(s|C_{I_k}) \leqslant \epsilon \tag{11}$$

for sufficiently large modulus $(p_0, p_1, \ldots, p_n)$, where $H(s)$ is the (Shannon) entropy of the secret s and $H(s|C_{I_k})$ is the conditional entropy of s given the k RCs C_{I_k}.

In this subsection we disprove the Claim 2. Our main result can be presented by Theorem 2 and Heuristic 2.

Theorem 2. *Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the CRT-based (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. For any modulus $(p_0, p_1, \ldots, p_n)$, there exists a k_0 such that, if $k \geqslant k_0$, then with high probability we have $\Delta(C_{I_k}) = H(s)$.*

Heuristic 2. Suppose there are m $(m \geqslant t)$ participants collaborating to reconstruct the secret s in the CRT-based (t, m, n) GOSS scheme. Let the RCs of the m participants be $C_{I_m} = (c_{i_1}, \ldots, c_{i_m})$. Suppose the unauthenticated adversary $\mathscr{A}$ knows a subset of k RCs, among the $m, C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. For any $k \geqslant t$, we have $\Delta(C_{I_k}) > \mathrm{negl}(p_0)$.

Similarly, Theorem 2 and Heuristic 2 actually say that there exists an unauthenticated adversary $\mathscr{A}$ who can join the reconstruction phase of the CRT-based GOSS scheme and successfully obtain the secret s when more than t honest shareholders participant in the reconstruction phase. That is the CRT-based GOSS scheme provides basically no better security against unauthenticated outside adversary than plain Asmuth-Bloom's threshold scheme.

Assume m users $U_{I_m} = (U_{i_1}, \ldots, U_{i_m})$ present at the reconstruction phase. Now we present the attack strategy for an outside adversary $\mathscr{A}$ who knows k $(k > t)$ RCs $C_{I_k} = (c_{i_1}, \ldots, c_{i_k})$. Denote the k authentic users, whose RCs are observed by the adversary $\mathscr{A}$, as $U_{I_k} = (U_{i_1}, \ldots, U_{i_k})$. Note that the adversary knows public information of all the m users as $\left(p_{i_1}, \ldots, p_{i_m}\right)$. Denote the shares of U_{I_k} as $S_{I_k} = (s_{i_1}, \ldots, s_{i_k})$.

Recall that the share for U_{i_j} where $i_j \in I_k$ is $s_{i_j} = s + \alpha p_0 \bmod p_{i_j}$. Set $s' = s + \alpha p_0$. When m shareholders collaborate to reconstruct the secret, the randomized components are computed as $c_{i_j} = \left(s_{i_j} \frac{N}{p_{i_j}} y_{i_j} + r_{i_j} \frac{N}{p_{i_j}} p_0\right) \bmod N$, where $N = \prod_{i_j \in I_m} p_{i_j}$ and $y_{i_j} = \left(N/p_{i_j}\right)^{-1} \bmod p_{i_j}$. Set $N/p_{i_j} = N_{i_j}$. Since for every $i_j \in I_m, N_{i_j} | s_{i_j} N_{i_j} y_{i_j}, N_{i_j} | r_{i_j} N_{i_j} p_0$ and $N_{i_j} | N$, we have $N_{i_j} | c_{i_j}$. Let $c'_{i_j} = c_{i_j}/N_{i_j}$, we further have

$$c_{i_j}/N_{i_j} = \left(s_{i_j} y_{i_j} + r_{i_j} p_0\right) \bmod p_{i_j}. \tag{12}$$

Since $s_{i_j} = s' \mathrm{mod} p_{i_j}$, we have

$$c_{i_j}/N_{i_j} = \left(s' y_{i_j} + r_{i_j} p_0\right) \mathrm{mod}\ p_{i_j}. \tag{13}$$

Multiply both sides of Eq. (13) by N_{i_j}, we get

$$c_{ij} = s' + r_{i_j} p_0 N_{i_j} \bmod p_{i_j}. \tag{14}$$

Rearrange Eq. (14) we get the following equation.

$$s' = c_{ij} - r_{i_j} p_0 N_{i_j} \bmod p_{i_j}. \tag{15}$$

Now the adversary $\mathscr{A}$ gets access to k RCs, thus she can obtain the following system of modular equations of size k.

$$\begin{cases} s' = c_{i_1} - \quad r_{i_1} p_0 N_{i_1} \bmod p_{i_1}; \\ \qquad\qquad \cdots \\ s' = c_{i_k} - \quad r_{i_k} p_0 N_{i_k} \mathrm{mod}\ p_{i_k}. \end{cases} \tag{16}$$

The goal of the adversary $\mathscr{A}$ is to try to compute s' from the equation system Eq. (16) and then obtain the secret $s = s' \bmod p_0$. The equation system Eq. (16) is a modulo equation system with all modulus co-prime and can be solved by

CRT. However, unfortunately the adversary only knows c_{ij}. So she is not sure about the right side of the modular equation system due to the unknown random elements r_{ij}.

Here we provide a naive strategy for the adversary $\mathscr{A}$: she just enumerates all possible values of $(r_{i_1}, \cdots, r_{i_k}) \in \mathbb{Z}_{p_0}^k$ and solves Eq. (16) to get s'. In order to prove Theorem 2, we investigate the characterization of the solutions of Eq. (16). Note that the method of obtaining the solutions is irrelevant to the correctness of our proof. Precisely, the loss of entropy of the secret is not affected by the method to solve Eq. (16). We will prove that when the adversary sees enough RCs, then, with overwhelmingly large probability, Eq. (16) has a unique solution. Therefore, the adversary can uniquely determine the shared secret s. On the other hand, when the adversary can only obtain limited RCs, Eq. (16) might have more than one solutions. However, in most cases, those solutions do not exhibit a uniform distribution. This non-uniform distribution of solutions leaks information on the secret to the adversary. For example, some possible choices for the secret might not be solutions to Eq. (16). Then the adversary can exclude such values as the secret, which means the entropy of the secret is reduced due to the adversary's knowledge of RCs. Specifically, we prove that even in the case that the number of RCs the adversary can get is the threshold t, the loss of entropy of the secret is non-negligible.

Now we prove Theorem 2 and justify Heuristic 2.

Proof of Theorem 2. For clarity, we denote the true secret shared by the dealer by s^* and accordingly $s^{*\prime} = s^* + \alpha p_0$. Since there are p_0^k possible choices for $(r_{i_1}, \cdots, r_{i_k})$ the adversary gets p_0^k possible values of s' by solving Eq. (16) with CRT. According to CRT, all the possible values of s' belong to the ring $\mathbb{Z}_{N_k}$ with N_k defined as $N_k = \prod_{j=1}^k p_{i_j}$. Let $M = \lceil p_1 \cdot \ldots \cdot p_t / p_0 \rceil$. Recall that according to the **ShareGen** algorithm (cf. Fig. 2) of CRT-based GOSS, $s^{*\prime} \in \mathbb{Z}_M$. We assume, without loss of generality, that $p_{i_1} < p_{i_2} < \cdots < p_{i_k}$.

Let $N_u = \prod_{j=1}^u p_{i_j}$ and denote by S_u the set of solutions those satisfy the first $u (1 \leqslant u \leqslant k)$ equations of Eq. (16) and are less than M. Specifically,

$$S_u = \left\{ s_u \middle| \left(s_u = c_{i_j} - r_{r_j} \bmod p_{i_j} \text{for } j = 1, \cdots, u \right) \wedge s_u < M \right\}.$$

The adversary $\mathscr{A}$ can enumerate all p_0^u possible values of the random elements $(r_{i_1}, \cdots, r_{i_u})$. And each combination of $(r_{i_1}, \cdots, r_{i_u})$ values gives a unique solution of $s' \leftarrow \mathbb{Z}_{N_u}$. When $N_u < M$, each possible solution $s' \in \mathbb{Z}_{N_u}$ corresponds to roughly M/N_u ($\lceil M/N_u \rceil$ or $\lceil M/N_u \rceil - 1$) solutions in $\mathbb{Z}_M$, in the form of $s' + iN_u$, where $i = 0, 1, \cdots, \lfloor M/N_u \rfloor$. So the number of occurrences of candidate solutions (those s' which belong to $\mathbb{Z}_M$) is roughly $|S_u| = p_0^u M/N_u$. For convince, we will use M/N_u instead of $\lfloor M/N_u \rfloor$ throughout this proof.

When $N_u > M$, for each possible choice of $(r_{i_1}, \cdots, r_{i_u})$, the unique solution $s' \in \mathbb{Z}_{N_u}$ corresponds to at most one solution in $\mathbb{Z}_M$. Hence, there are at most p_0^u possible solutions in $\mathbb{Z}_M$. The next question is how many of the p_0^u possible solutions in $\mathbb{Z}_{N_u}$ are actually in $\mathbb{Z}_M$. Set $v = \text{argmax}_v M > N_v = \prod_{j=1}^v p_{i_j}$. According to the above analysis, we have $|S_v| = p_0^v M/N_v$. And we can express S_v as $S_v = \{ s_v | s_v = s_i' + jN_v \}$, where $i = 1, \cdots, p_0^v$ and $j = 0, \cdots, M/N_v$. For any fixed $i \in [p_0^v]$, we define $S_{v,i} = \{ s_i' + jN_v \} \subset S_v$, where $j = 0, \cdots, M/N_v$. Since $N_{v+1} = N_v p_{i_{v+1}} > M$, we have $p_{i_{v+1}} > M/N_v$. Certainly, $S_{v+1} \subset S_v$. Now we consider the set $S_{v,i} \cap S_{v+1}$ for a fixed s_i'. Note that $|S_{v,i}| = M/N_v$. It is easy to see that for any 2 different elements $s_i' + j_1 N_v$ and $s_i' + j_2 N_v$ belonging to $S_{v,i}$, we have $s_i' + j_1 N_v \neq s_i' + j_2 N_v \bmod p_{i_{v+1}}$. Thus, there are M/N_v different values modulo $p_{i_{v+1}}$ in $S_{v,i}$. While according to the choice of $r_{i_{v+1}}$, at most p_0 elements in $S_{v,i}$ belong to S_{v+1}. However, since $M/N_v < p_{i_{v+1}}, |S_{v,i} \cap S_{v+1}| \leqslant p_0$. We assume that the p_0 possible values of j such that $s_i' + j_1 N_v$ satisfies the $s_i' + j_1 N_v = c_{i_{v+1} - r_{i_{v+1}}}$ are uniformly distributed in $\left[p_{i_{v+1}} \right]$. Then the expected number of $|S_{v,i} \cap S_{v+1}|$ is

$$\mathbb{E}\left(|S_{v,i} \cap S_{v+1}|\right) = \sum_{j=0}^{p_0} \frac{j \binom{p_{i_{v+1}} - j}{M/N_v - j} \binom{p_0}{j}}{\binom{p_{i_{v+1}}}{M/N_v}}.$$

So the expected size of S_{v+1} is

$$\mathbb{E}(|S_{v+1}|) = p_0^v \sum_{j=0}^{p_0} \frac{j \binom{p_{i_{v+1}} - j}{M/N_v - j} \binom{p_0}{j}}{\binom{p_{i_{v+1}}}{M/N_v}}.$$

We do not need to estimate $\mathbb{E}(|S_{v+1}|)$. Just notice that $\mathbb{E}(|S_{v+1}|) < \mathbb{E}(|S_v|)$. We can view the above analysis as a seiving process to solve the CRT equation system. Similarly we get $\mathbb{E}(|S_{v+2}|) < \mathbb{E}(|S_{v+1}|)$. Finally, we conclude that when k is large enough, $\mathbb{E}(|S_k|) = 1$. So there exists a k_0 such that, for any $k \geqslant k_0$ with high probability there is only one possible value of s' which satisfies $s' \in \mathbb{Z}_M$ and it is easy to see that $s' = s^{*\prime}$ since $s^{*\prime}$ is a solution to the equation. Hence, the adversary can get the true secret s by calculating $s = s' \bmod p_0$. Equivalently, we say $H(s|C_{I_k}) = 0$ with high probability if $k \geqslant k_0$, i.e., $\Delta(C_{I_k}) = H(s)$.

Justification of Heuristic 2. Without loss of generality, we only argue for the case $k = t$. When $k = t$ it is highly likely that $|S_k| > 1$. Now the adversary $\mathscr{A}$ can not uniquely determine s' hence the secret s. But she can however record all possible

$s' \in \mathbb{Z}_M$ and the according secret $s' \bmod p_0 = s \in \mathbb{Z}_{p_0}$. Same as above, we set $v = \text{argmax}_v M > N_v = \prod_{j=1}^{v} p_{i_j}$. Note that v must exist and $v < k$, since when $k = t$ we have $N_k > M$. For the set S_v, we have $S_v = \{s_v | s_v = s'_i + jN_v\}$, where $i = 1, \cdots, p_0^v$ and $j = 0, \cdots, M/N_v$. Thus, the set $S_{v,i} = \{s'_i + jN_v\}$ for fixed i is almost uniformly distributed over $\mathbb{Z}_{p_0}$ since for every consecutive p_0 choices of j, jN_v traverse $\mathbb{Z}_{p_0}$. However, for the set S_{v+1}, its elements are distinct modulo M, the above property does not hold anymore. Thus, the elements in S_{v+1} are not uniformly distributed in $\mathbb{Z}_{p_0}$. We find it difficult to bound the statistical distance between $S_k \bmod p_0$ and uniform distribution over $\mathbb{Z}_{p_0}$. So we present the second statement that for any $k \geqslant t, \Delta(C_{I_k}) = H(s) - H(s|C_{I_k}) > \text{negl}(p_0)$ as a heuristic.

6. Experimental Results

In this section we provide experimental results on our attack methods against the polynomial-based GOSS scheme and CRT-based GOSS scheme. All our attacks are implemented using SageMath [31] script and the codes are freely available online[3]. Since our Theorem 1 and Theorem 2 prove that the two GOSS schemes are not information-theoretically secure and we are considering a computation unbounded adversary, we illustrate our attacks on small parameters.

6.1. Experimental Results of Attacks Against Polynomial-based GOSS

We choose a set of parameters for the GOSS scheme as follows. There are $n = 10$ users and the threshold is $t = 4$. The secret s belongs to the filed $\mathbb{F}_5$, i.e., the prime number $q = 5$. The prime $p = 257$ which is the smallest prime number satisfying the relation $p > q + nq^2$. The public information of a user U_i is simply set as $x_i = i \in \mathbb{F}_p$. The parameters we use for GOSS are listed in Table 1.

Firstly, we examine the attack in the case $k > t$ with k being the number of RCs observed by the adversary $\mathscr{A}$. The k RCs are taken from m RCs. For simplicity, we set $m = k + 1$ and take the first k RCs (i.e., C_{I_k}) as being observed by the adversary $\mathscr{A}$. After obtaining the k RCs, we construct the matrix $\mathbf{A}$, the vector $\mathbf{c}$ as described by Eq. (8) and solve the LWE problem using the package fpylll, which is a Python library for performing lattice reduction on lattices over the integers based on the C++ library fplll [32]. In the experiment, we first choose a secret $s \xleftarrow{\$} \mathbb{F}_q$ at random. Then for each choice of s we randomly generate n shares, and select $m = k + 1$ users randomly. Finally we generate m RCs using the shares of the m users and take the first k RCs to construct the LWE problem. For each value of $k \in [5, 6, 7, 8, 9]$, we repeat the above process 1000 times and record the number of times that the adversary can correctly recover the original secret shared by the dealer. This process is again repeated 100 times to estimate the average success probability of the adversary. Fig. 3 plots the mean and standard deviation of the successful probability of the adversary for various choices of k, which is the number of RCs the adversary can observe. .

Secondly, we examine the attack in the case $k = t$. Similarly, we set $m = k + 1$ and take the first k RCs (i.e., C_{I_k}) as those being observed by the adversary $\mathscr{A}$. In the experiment, we first choose a secret $s \xleftarrow{\$} \mathbb{F}_q$ at random. Then for each choice of s we randomly generate n shares, and select $m = k + 1$ users randomly. Finally we generate m RCs using the shares of the m users and take the first k RCs to construct the modulo linear system as described by Eq. (9). For each choice of s we enumerate the vector $\mathbf{r} \in \mathbb{F}_q^k$ and solve Eq. (9) to get the secret s. The above process is repeated 1000 times. According to the experimental data, we find that even though in some trials, the adversary cannot uniquely determine the shared secret, the candidates are fewer than q.

For example, in one trial (please refer to the first line of Table 2) when the secret was sampled as $s = 2$, after enumerating all q^k possibilities of $\mathbf{r}$, the adversary finds that for 2 possible values of $\mathbf{r}$, the corresponding secret solved from Eq. (9) is $s = 4$ which belongs to $\mathbb{F}_q$; and for 10 possible values of $\mathbf{r}$, the corresponding secret solved from Eq. (9) is $s = 2$ which belongs to $\mathbb{F}_q$. None of the q^k possibilities of $\mathbf{r}$ can result a secret s of 0 or 3 or 4. In this case, even if the adversary cannot uniquely determine the original secret, the entropy of the secret decreases substantially.

In the experiment, we also record all the failure cases when the adversary cannot uniquely determine the secret. In such cases, we record the *candidate secret vector* which is a vector of size q with its i-th element denoting the number of occurrence that $s = i \in \mathbb{F}_q$ satisfies Eq. (9). For example, in the instance above the candidate vector is $(0, 0, 10, 0, 2)$. Refer to Table 2 for a selected exhibition of the distribution of candidate vectors in the failure cases. Since we have recorded the number of success trials and the distribution of candidate vectors for the failure cases, now we can calculate the empirical conditional entropy $\widehat{H}(s|C_{I_k})$. According to the experiments data we calculate the empirical conditional entropy as $\widehat{H}(s|C_{I_k}) = 0.375$, which is quite small. On the other hand, the entropy of the secret is $H(s) = \log q = \log 5 = 2.32$. Hence, the claim in our Theorem 1 is supported by the experiments. In the failure cases, although the adversary cannot uniquely determine the true secret that was shared by the dealer, she can observe the distribution of candidate vectors. We take a naive strategy for the adversary to guess the original secret as s' where s' is determined by maximum likelihood estimation.

Table 2 also shows the adversary's guess and whether the guess is successful or not. In our experiment when $k = t$, there are 223 trials out of 1000 in which cases the adversary cannot uniquely determine the shared secret. But using the maximum

[3] https://github.com/little-worm/Shamir-Goss

Table 1
Parameters for polynomial-based GOSS

#users: n	threshold: t	secret size: q	share size: p
10	4	5	257

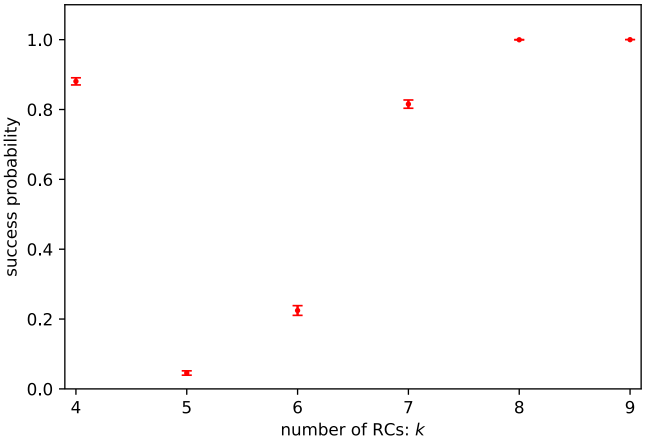


Fig. 3. Mean and standard deviation of adversary's success probability against polynomial-based GOSS.

Table 2
Selected exhibition of candidate secret vectors distribution in failure cases of polynomial-based GOSS when $k = t$

secret: s	candidate vector	guess secret: s'	guess successful
2	(0, 0, 10, 0, 2)	2	yes
0	(24, 20, 19, 18, 16)	0	yes
0	(29, 30, 27, 28, 25)	1	no
3	(27, 28, 25, 24, 21)	0	no
4	(0, 0, 1, 0, 2)	4	yes

likelihood estimation the adversary can correctly guess the shared secret in 108 out of the 223 failure trials. We also repeated the above process 100 times and plot the mean and deviation of the adversary's success probability in Fig. 3. The success probability also counts the cases when there are no unique solutions but the adversary correctly guesses the original secret using maximum likelihood estimation.

From Fig. 3 we can see that the adversary's average success probability is 85% when she obtains t RCs. The figure also reveals the fact that when $k > t$ the adversary's success probability increases quite fast. For example adversary's average success probability is 4% when $k = 5$ but it increases to 99% when $k = 8$. Moreover in our experiments, in each of the 100 trials our attack successfully recovers the original secret when the adversary obtains $k = 9$ RCs with 100% success probability. Note that the average success probability drops quite substantially from $k = 4$ to $k = 5$. This is because in the case of $k = 5$ we attack the scheme by solving an LWE instance. When k is relatively small (i.e., close to t), the LWE problem might not have a unique solution. Nevertheless, the dropdown of success probability is rather artificial since in the case when $k > t$ we can still use the enumeration strategy to attack the scheme and get higher success probability. We choose not to explore on this side further because it is obvious. The reason is that the enumeration strategy works fine with $k = t = 4$, it certainly works better when $k > t$. Moreover, our results of $k > t$ show that when k is relatively large, the adversary can use sophisticated LWE solving techniques, such as enumeration with extreme pruning [27], unique-SVP [33] and Coded-BKW [34], to speedup her attack.

Table 3
Successful probability of adversary against polynomial-based $(20, 100, k + 1)$ GOSS

k	35	36	37	38	39	40
$\Pr[\mathscr{A}\text{wins}]$	74.2%	87%	95.1%	97.4%	99.4%	100%

Table 4
Number of available RCs enabling 100% success probability of adversary against $(t, 100, k+1)$ GOSS

threshold: t	20	25	30	35	40
# of RCs: k	40	51	65	75	95

Table 5
Parameters for CRT-based GOSS

n	t	p_0	$(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10})$
10	4	5	$(313, 317, 331, 337, 347, 349, 353, 359, 367, 373)$

Table 6
Selected exhibition of candidate secret vectors distribution in failure cases of CRT-based GOSS

k:	secret: s	candidate vector	guess: s'	guess successful
4	3	(8, 10, 30, 35, 28)	3	yes
4	4	(30, 28, 15, 33, 17)	3	no
4	1	(0, 44, 5, 21, 39)	1	yes
5	3	(0, 0, 2, 2, 0)	2	no
5	2	(0, 0, 2, 1, 0)	2	yes
5	2	(0, 2, 1, 0, 2)	1	no
6	1	(0, 1, 0, 0, 1)	1	yes
6	4	(1, 0, 0, 0, 1)	0	no
6	0	(1, 0, 0, 1, 0)	0	yes

Table 7
Conditional entropy of the secret and overall successful probability of adversary against CRT-based GOSS

k	$\widehat{H}(s\|C_{I_k})$		$\Pr[\mathscr{A}\text{wins}]$	
	mean	std	mean	std
4	1.89	0.012	37.68%	1.70%
5	0.90	0.018	62.17%	1.45%
6	0.0083	0.0033	99.57%	0.21%
7	0	0	100%	0
8	0	0	100%	0
9	0	0	100%	0

We conducted further examples to exhibit the effectiveness of modeling the adversary's attack as solving LWE variant. Firstly, we set the threshold to be $t = 20$, the number of users to be $n = 100$, and the secret size to be $q = 53$. Then $p = 280957$ is the smallest prime number satisfying the relation $p > q + nq^2$. For the case when the adversary observes $k \in [35, 36, 37, 38, 39, 40]$ RCs, we run the attack algorithm by solving the corresponding LWE problem, for 1000 times. We list the successful probability of the adversary in Table 3. Secondly, we fix $n = 100, q = 53$ and $p = 280957$ while change the threshold t to check k, the number of available RCs, which enable the adversary to recover the shared secret in polynomial-based $(t, n, k+1)$ GOSS with 100% successful probability. The values of k's are recorded in Table 4.

6.2. Experimental Results of Attacks Against CRT-based GOSS

We choose a set of parameters for the polynomial-based GOSS scheme as follows. There are $n = 10$ users and the threshold is $t = 4$. The secret s belongs to the filed $\mathbb{F}_5$, i.e., the prime number $p_0 = 5$. The prime numbers p_i for $i \in [n]$ are consecutive primes larger than p_0 and we have verified that they satisfy the relation $p_0^2 \cdot p_{n-t+2} \cdots p_n < p_1 p_2 \cdots p_t$. The public information of a user U_i is simply set as $x_i = i \in \mathbb{F}_{p_i}$. The parameters we use for polynomial-based GOSS are listed in Table 5.

We first choose a secret $s \xleftarrow{\$} \mathbb{Z}_{p_0}$ at random and generate n shares. Then we randomly generate $m = k + 1$ RCs using the shares of m randomly selected users and take the first k RCs (i.e., C_{I_k}) as those observed by the adversary $\mathscr{A}$. We use the k RCs to construct a linear system of modulo equations as described by Eq. (16). For each choice of s we enumerate the vector $(r_{i_1}, \ldots, r_{i_k}) \in \mathbb{Z}_{p_0}^k$ and solve Eq. (16) to get a solution s'. If $s' \in \mathbb{Z}_M$ (refer to the proof of Theorem 2), then $s' \bmod p_0$ is a possible candidate of the original secret s shared by the dealer. There are two cases:

1. For all the p_0^k possible values of $(r_{i_1}, \ldots, r_{i_k})$ the solution s' modulo p_0 is unique. In this case, the adversary can successfully recover the original secret s.
2. Enumerating over all p_0^k possible values, the solutions s' modulo p_0 are not unique (of course, we have excluded those s''s which do not belong to $\mathbb{Z}_M$). In this case, we record the candidate secret vector whose i-th element denotes the number of occurrence that s' is a solution of Eq. (16) and $s' = i \bmod p_0$.

For each value of $k \in [4, 5, 6, 7, 8, 9]$, we repeat the above process 1000 times and record the experimental data. Same as above we exhibit some of the failure cases when the adversary cannot uniquely determine the secret in Table 6.

From Table 6 we can see that even if the adversary $\mathscr{A}$ cannot uniquely determine the shared secret, the candidate secrets are usually a subset of $\mathbb{F}_{p_0}$ and she has a large chance of success if choosing to guess the secret by maximum likelihood estimation. Further we can observe that when $k = t$ ($k = 4$ in our example) is the threshold, the candidate vector is not sparse. That means at least to the adversary every secret $s \in \mathbb{F}_{p_0}$ is a possible candidate. But the crucial point here is that the candidate vector is not uniformly distributed, the adversary gets information about the real secret. Moreover, as we will show in the sequential that using the maximum likelihood estimate, the adversary successfully guess the correct secret with probability noticeably better than a random guess. From Table 6, the candidate vector gets sparse when k increases. In our experiments no failure case occurred when $k \geqslant 7$. Similarly we calculate the empirical conditional entropy $\widehat{H}(s|C_{I_k})$ and the overall successful probability of the adversary $\Pr[\mathscr{A}\text{wins}]$ according to the experiments data and list them in Table 7. Note that the entropy of the secret s is $H(s) = \log p_0 = \log 5 = 2.32$. From Table 7 we can see that our attack method works well. The probability that the adversary successfully recovers the shared secret increases rapidly when she gets more RCs. Even that the adversary $\mathscr{A}$ only observes t RCs, her successful probability is around 37.68% which is much higher than a random guess (20% in the case $p_0 = 5$). In our experiments when $\mathscr{A}$ observes 7 RCs or more, she always succeeds in recovering the shared secret.

7. Conclusion

The original definition of (t, n) threshold secret sharing scheme does not prevent an outside adversary who has no share but participate in the reconstruction phase from getting the shared secret. Although this problem can be resolved simply using user authentication, the overhead would be $O(n^2)$. Thus, researchers proposed a notion called group oriented secret sharing which tries to capture the issue of unauthenticated users in threshold schemes. Basically the group oriented property guarantees that even if more than t shareholders participate in the reconstruction phase, the secret can be recovered correctly only if all the participants are valid shareholders.

In this paper we focus on two GOSS schemes proposed by Miao et al. [15,16] and show that these two schemes are not as secure as claimed. We develop brute-force-style attack methods by which the outside adversary can recover the original secret shared by the dealer with high probability if she obtains t or more RCs from honest shareholders. Moreover, for the polynomial-based GOSS our attack method can be modeled as solving an LWE instance which has faster algorithm than enumeration.

It seems that authentication might be the most natural tool to deal with the outside adversary problem. It would be interesting to investigate whether any secure GOSS scheme implies an authentication scheme.

As a side note, we point out that some other variants of secret sharing schemes, such as verifiable secret sharing [9] and cheater identifiable secret sharing [13], can effectively solve the problem of defending against unauthenticated outside adversary.

CRediT authorship contribution statement

Rui Xu: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing, Funding acquisition. **Xu Wang:** Methodology, Validation, Visualization. **Kirill Morozov:** Conceptualization, Validation, Writing - review & editing. **Chi Cheng:** Conceptualization, Validation, Writing - review & editing, Funding acquisition. **Jintai Ding:** Methodology, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work presented in this paper was supported in part by the National Natural Science Foundation of China under Grant No. 61802354 and 62172374, and Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information

Processing under Grant No. KLIGIP 2021B07. We highly appreciate the anonymous reviewers for their critical assessment and suggestive comments of our work, which helped us to improve the quality of this article.

References

[1] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.
[2] G.R. Blakley, Safeguarding cryptographic keys, in: 1979 International Workshop on Managing Requirements Knowledge (MARK), IEEE, 1979, pp. 313–318.
[3] C. Asmuth, J. Bloom, A modular approach to key safeguarding, IEEE Transactions on Information Theory 29 (2) (1983) 208–210.
[4] Y.G. Desmedt, Threshold cryptography, European Transactions on Telecommunications 5 (4) (1994) 449–458.
[5] T. Rabin, M. Ben-Or, Verifiable secret sharing and multiparty protocols with honest majority, in: Proceedings of the twenty-first annual ACM symposium on Theory of computing, 1989, pp. 73–85.
[6] R. Cramer, I. Damgård, U. Maurer, General secure multi-party computation from any linear secret-sharing scheme, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2000, pp. 316–334.
[7] P. Singh, B. Raman, Reversible data hiding based on shamirs secret sharing for color images over cloud, Information Sciences 422 (2018) 77–97.
[8] L. Harn, Secure secret reconstruction and multi-secret sharing schemes with unconditional security, Security and Communication Networks 7 (3) (2014) 567–573.
[9] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: Annual international cryptology conference, Springer, 1991, pp. 129–140.
[10] B. Rajabi, Z. Eslami, A verifiable threshold secret sharing scheme based on lattices, Information Sciences 501 (2019) 655–661.
[11] K.M. Martin, J. Pieprzyk, R. Safavi-Naini, H. Wang, Changing thresholds in the absence of secure channels, in: in: Australasian Conference on Information Security and Privacy, Springer, 1999, pp. 177–191.
[12] X. Jia, D. Wang, D. Nie, X. Luo, J.Z. Sun, A new threshold changeable secret sharing scheme based on the chinese remainder theorem, Information Sciences 473 (2019) 13–30.
[13] R. Xu, K. Morozov, T. Takagi, Cheater identifiable secret sharing schemes via multi-receiver authentication, in: International Workshop on Security, Springer, 2014, pp. 72–87.
[14] Z. Ahmadian, S. Jamshidpour, Linear subspace cryptanalysis of harn's secret sharing-based group authentication scheme, IEEE Transactions on Information Forensics and Security 13 (2) (2017) 502–510.
[15] F. Miao, Y. Xiong, X. Wang, B. Moaman, Randomized component and its application to (t, m, n)-group oriented secret sharing, IEEE Transactions on Information Forensics and Security 10 (5) (2015) 889–899.
[16] F. Miao, Y. Fan, X. Wang, Y. Xiong, M. Badawy, A (t, m, n)-group oriented secret sharing scheme, Chinese Journal of Electronics 25 (1) (2016) 174–178.
[17] L. Harn, Group authentication, IEEE Transactions on computers 62 (9) (2012) 1893–1898.
[18] H.-Y. Chien, Group authentication with multiple trials and multiple authentications, Security and Communication Networks 2017 (2017) 1–7, https://www.hindawi.com/journals/scn/2017/3109624/.
[19] D.-H. Lee, I.-Y. Lee, Dynamic group authentication and key exchange scheme based on threshold secret sharing for iot smart metering environments, Sensors 18 (10) (2018) 3534.
[20] Y. Aydin, G.K. Kurt, E. Ozdemir, H. Yanikomeroglu, A flexible and lightweight group authentication scheme, IEEE Internet of Things Journal 7 (10) (2020) 10277–10287.
[21] Z. Xia, Y. Liu, C.-F. Hsu, C.-C. Chang, Cryptanalysis and improvement of a group authentication scheme with multiple trials and multiple authentications, Security and Communication Networks (2020).
[22] R. Xu, X. Wang, K. Morozov, Group authentication for cloud-to-things computing: Review and improvement, Computer Networks 108374 (2021).
[23] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM) 56 (6) (2009) 34.
[24] D. Micciancio, Lattice-based cryptography, Encyclopedia of Cryptography and Security (2011) 713–715.
[25] L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky, Lattice signatures and bimodal gaussians, in: Annual Cryptology Conference, Springer, 2013, pp. 40–56.
[26] M.R. Albrecht, R. Player, S. Scott, On the concrete hardness of learning with errors, Journal of Mathematical Cryptology 9 (3) (2015) 169–203.
[27] M. Liu, P.Q. Nguyen, Solving bdd by enumeration: An update, in: Cryptographers's Track at the RSA Conference, Springer, 2013, pp. 293–309.
[28] R. Xu, S.L. Yeo, K. Fukushima, T. Takagi, H. Seo, S. Kiyomoto, M. Henricksen, An experimental study of the bdd approach for the search lwe problem, in: D. Gollmann, A. Miyaji, H. Kikuchi (Eds.), Applied Cryptography and Network Security, Springer International Publishing, Cham, 2017, pp. 253–272.
[29] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, D. Stebila, Frodo: Take off the ring! practical, quantum-secure key exchange from lwe, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1006–1018.
[30] M. Quisquater, B. Preneel, J. Vandewalle, On the security of the threshold scheme based on the chinese remainder theorem, in: International Workshop on Public Key Cryptography, Springer, 2002, pp. 199–210.
[31] The Sage Developers, SageMath, the Sage Mathematics Software System (Version 9.1), https://www.sagemath.org (2020).
[32] The FPLLL Development Team, Fplll, a lattice reduction library, available at https://github.com/fplll/fplll (2020).
[33] M.R. Albrecht, R. Fitzpatrick, F. Göpfert, On the efficacy of solving lwe by reduction to unique-svp, in: International Conference on Information Security and Cryptology, Springer, 2013, pp. 293–310.
[34] Q. Guo, T. Johansson, P. Stankovski, Coded-BKW: Solving LWE using lattice codes, in: Annual Cryptology Conference, Springer, 2015, pp. 23–42.