


A New Gradual Secret Sharing Scheme with Diverse Access Structure

Jamal Zarepour-Ahmadabadi¹ · MohammadEbrahim Shiri-Ahmadabadi¹  · Ali Miri² · AliMohammad Latif³

Published online: 8 January 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract A multi-secret sharing scheme tries to share multiple secrets among a group of players in such a way that co-operation of pre-specified subsets of them, called *access structures*, can reconstruct the secrets. Existing methods allow for all secrets to be recovered at once, by the same sharing algorithm, and by identical access structures. However, in many real world applications, secrets may not needed all at once, access structure may vary for different secrets (change over time), and a group of dishonest players may collude to obtain all secrets. In this paper, we propose a novel and efficient algorithm to address these issues. Our main objectives are, to recover each secret according to its own scheme, by its own access structure, and whenever needed. Our proposed algorithm also blocks collusion attacks by dishonest players. Our scheme can work with any general purpose threshold schemes. It is also rather efficient in terms of computational and communication overhead costs. There computational costs for sharing and recovering stages are almost negligible, and communication costs of sharing and recovering are of order $\mathcal{O}(n+k)$ and $\mathcal{O}(\sum_{i=1}^k t^i)$ respectively, where n is the number of players and where t^i 's are the threshold values for the k secrets.

✉ MohammadEbrahim Shiri-Ahmadabadi
Shiri@aut.ac.ir

Jamal Zarepour-Ahmadabadi
zarepourjamal@gmail.com

Ali Miri
Ali.Miri@ryerson.ca

AliMohammad Latif
alatif@yazd.ac.ir

¹ Department of Computer Science, Faculty of Mathematics and Computer Science, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

² Department of Computer Science, Ryerson University, Toronto, Canada

³ Department of Computer Engineering, Yazd University, Yazd, Iran

Keywords Cryptography · Threshold scheme · Chinese remainder theorem · Multi-secret sharing scheme · Trusted third party (TTP)

1 Introduction

One of the most important challenges in the era of information technology is protecting privacy of information. *Secret Sharing (SS)* schemes address this issue in group settings. SS schemes protect the secrets both in terms of unauthorized access, and from being lost or corrupted. Some of the earliest on design of SS schemes can be found in work of Shamir [21] and Blackely [2], and many extensions and refinements can be found in the literature [1, 6, 8, 14, 23, 24].

The most commonly used algorithms, in existing SS schemes, use a (t, n) -threshold approach, in which a secret is distributed among a group of n parties in such a way that the secret can be reconstructed by having at least t shares. These schemes can share only one secret per execution, and there is no control over the honesty of parties.

To tackle the malicious behaviour of dishonest participants, some SS schemes offer a *verifiability* option. In a verifiable secret sharing scheme, the validity of the shares can be verified by computing a one-way function and broadcasting public information.

In many real world scenarios, it is common that settings have multiple separated parts, where each part has its own secret key. *Multi-secret sharing (MSS)* schemes [4, 6, 8–10, 13, 16, 19, 22, 25] were introduced to address these kind of scenarios. These MSS schemes allow for sharing of multiple secrets, that are recovered simultaneously in the reconstruction phase. However, these schemes also lack control over the reconstruction process, as they assume a homogeneous setup in terms of access structure, and the release of secrets. To illustrate this lack of control, consider an example of an online game with k levels, in which players have to collaborate to complete different levels of the game. To participate in a given level, a player not only has to successfully complete, with acceptable performance in the previous level, but also be part of and collaborate with a pre-specified, i.e threshold number of players who have reached that level. These kinds of examples cannot be dealt with existing MSS schemes as:

1. All the secrets are recovered at once. In fact, one can treat the multiple secrets as one, by concatenating them!
2. Recovering all secrets at first increases the risk of misuse. For example in the case of the collaborative online game above, a player can ignore the pre-specified order of the game levels, or the work needed to complete each in order to advance to the next level.
3. The existing MSS schemes do not attempt to detect collusions among dishonest participants.
4. All secrets are shared in the same access structure. Furthermore, some algorithms [8, 20, 22] assume same threshold values for all secrets.

In this paper, a new method, called the *Gradual Secret Sharing (GSS)* scheme, is proposed to share a number of secrets among a set of participants such that:

- Each secret is recovered whenever it is needed;
- Each secret has its own sharing platform and access structure;
- Any number of colluding participants cannot obtain keys to all stages of the systems.

In our scheme, we use the *Chinese Remainder Theorem (CRT)* to bind the shares to each other. Then, the binded shares of the previous secret is used in the sharing platform of the

next secret. This process continues until all the secrets are shared. Later in the reconstruction phase, the first secret is recovered by co-operation of any pre-specified subset of participants. As participants successfully complete what is required for a given level, a trusted third party (TTP) with its own veto, referred here as an *administrator* will enable these participants to engage in secret sharing for the next level, if asked. This procedure will continue until all shares are recovered, or if the administrator stops the process. Note that unlike other CRT based schemes [1, 12], we use CRT as a binding tool.

The rest of the paper is organized as follows. In Sect. 2, basic notation, the CRT, and some more related papers are reviewed. The details of the proposed scheme are presented in Sect. 3. Section 4 analyzes the proposed scheme and then Sect. 5 compares it with similar schemes. Finally, conclusions are given in Sect. 6.

2 Preliminaries

In this section, first the symbols and notations that are used throughout the paper are listed. Then, some fundamental background of the proposed scheme are briefly reviewed.

2.1 Notations

Throughout this paper, we use the following notation, listed in the table below to describe our scheme.

D	The dealer
k	The number of secrets
SC^1, \dots, SC^k	The secrets to be shared
p	A sufficiently large prime number
$\mathfrak{A} = (\Gamma_1, \Gamma_2, \dots, \Gamma_k)$	Access structures corresponding to each secret
V^2, \dots, V^k	The manager's veto rights for each secrets
P_1, \dots, P_n	The participants
SH_1, \dots, SH_n	The shares to be distributed among participants
m_1, \dots, m_n	Pairwise relatively prime moduli of CRT

2.2 The Chinese Remainder Theorem

In number theory, the CRT states that, given the remainders of the division of an unknown integer Y by several integers that are co-prime, then one can uniquely determine Y , modulo by the product of these integers.

Theorem 1 *Let m_1, \dots, m_l be pairwise co-prime (that is $\gcd(m_i, m_j) = 1$ whenever $i \neq j$) and r_1, \dots, r_l are integers such that $r_i \in \mathbb{Z}_{m_i}$, then the system of l equations*

$$\begin{cases} Y \equiv r_1 \pmod{m_1}, \\ Y \equiv r_2 \pmod{m_2}, \\ \vdots \\ Y \equiv r_l \pmod{m_l}. \end{cases} \tag{1}$$

has a unique solution for Y modulo M , where $M = \prod_{i=1}^l m_i$ [3].

Define $b_i = M/m_i$ (the product of all the moduli except for m_i), and $b'_i = b_i^{-1} \pmod{m_i}$. Then,

$$Y = \sum_{i=1}^l r_i b_i b'_i \pmod{m_i} \tag{2}$$

is the unique solution.

Although, CRT has been widely used in SS schemes [1, 12, 17], we use it for binding the shares, and not for sharing/recovery process.

2.3 Shao’s Multi-Secret Sharing Scheme

To share k secrets in a (t, n) -threshold scheme, Shao [22] proposed a polynomial-based (k, t, n) multi-secret sharing scheme. Below, we briefly describe the two phases of this scheme—the *sharing*, and the *recovery* algorithms:

2.3.1 Sharing Algorithm

Depending on the values of k and t , the sharing algorithm executes one of the following cases:

- $t \geq k$

✓ D generates two polynomials of degree $t - 1$ as follows:

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \pmod{P},$$

$$g(x) = \sum_{i=0}^{t-1} b_i x^i \pmod{P},$$

where $a_0 = SC^1, \dots, a_{k-1} = SC^k$, and $a_k, \dots, a_{t-1}, b_0, \dots, b_{t-1}$ are random numbers from \mathbb{Z}_P^* and P is sufficiently large prime number.

✓ D calculates and publishes $v_i = H(f(i)||g(i))$ for $i = 1, 2, \dots, n$ and $c_i = b_i + ra_i \pmod{P}$ for $i = 0, 1, \dots, t - 1$, where $r = H(v_1||v_2||\dots||v_n)$ and $||$ is the concatenation sign.

✓ Each participant P_i receives his/her share $SH_i = (f(i), g(i))$, which can verify it via $v_i \stackrel{?}{=} H(f(i)||g(i))$, and $g(i) + rf(i) \stackrel{?}{=} \sum_{j=0}^{t-1} c_j i^j \pmod{P}$.

- $t < k$

✓ D generates two polynomials of degree $k - 1$ as follows:

$$f(x) = \sum_{i=0}^{k-1} a_i x^i \pmod{P},$$

$$g(x) = \sum_{i=0}^{k-1} b_i x^i \pmod{P},$$

where $a_0 = SC^1, \dots, a_{k-1} = SC^k$, and b_0, \dots, b_{k-1} are random numbers from \mathbb{Z}_p^* and P is sufficiently large prime number.

- ✓ D publishes $f(i)$ and $g(i)$ for $i = 1, 2, \dots, k - t$.
- ✓ D computes and publishes $v_i = H(f(k - t + i) || g(k - t + i))$ for $i = 1, 2, \dots, n$ and $c_i = b_i + ra_i \pmod P$ for $i = 0, 1, \dots, t - 1$, where $r = H(v_1 || v_2 || \dots || v_n)$ and $||$ is used for concatenating elements.
- ✓ Each participant P_i receives his/her share $SH_i = (f(k - t + i), g(k - t + i))$, which can verify it via $v_i \stackrel{?}{=} H(f(k - t + i) || g(k - t + i))$, and $g(k - t + i) + rf(k - t + i) \stackrel{?}{=} \sum_{j=0}^{t-1} c_j(k - t + i)^j \pmod P$.

2.3.2 Recovering Algorithm

Without loss of generality, we assume that the participants P_1, P_2, \dots, P_t cooperate to recover the secrets using their shares. Similar to the sharing algorithm, recovering stage execute one of the following two cases:

- If $t \geq k$:
The shares $SH_i = (f(i), g(i))$ are verified using $v_i \stackrel{?}{=} H(f(i) || g(i))$. Then, the $t - 1$ degree polynomial $f(x)$ is reconstructed by Lagrange interpolation.
- If $t < k$:
The shares $SH_i = (f(k - t + i), g(k - t + i))$ are verified via $v_i \stackrel{?}{=} H(f(k - t + i) || g(k - t + i))$. Then, the validated shares and published points, $f(i)$ for $i = 1, 2, \dots, k - t$, are used to reconstruct the polynomial $f(x)$ by using Lagrange interpolation.

The k first coefficients of the reconstructed polynomial $f(x)$ represent the secrets.

2.4 A Brief Review of Mashhadi’s Scheme

One of the most recent multi-stage secret sharing (MSSS) schemes is presented in the paper [18]. It is referred to as Mashhadi’s scheme later in our paper. This is a multi-secret sharing approach in which a dealer uses Non-Homogeneous Linear Feedback Shift Registers (NHLFSR) to share multiple-secrets among a set of participants, in such a way that any qualified subset of them can recover the secrets stage-by-stage. The advantages of this scheme are: fewer public values, simple distribution, and various ways of reconstruction.

In Mashhadi’s scheme [18], secrets are recovered in a predetermined order and each secret has its own access structure. However, the paper considers threshold access structure for all secrets. It should be pointed out that the access structures are not completely arbitrary. They must $\Gamma_i \subseteq \Gamma_{i+1}$, for $i = 1, 2, \dots, k - 1$, and $1 < t^1 \leq t^2 \leq t^k \leq n$, where t^i 's are threshold values for secrets SC^1, SC^2, \dots, SC^k , respectively.

The scheme is defined as $\Omega = (Setup, Distributin, Reconstruction)$ where:

1. *Setup* D selects a two-variable one-way function $f(r, s): \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}_p$. D also chooses n shares $s_1, s_2, \dots, s_n, s_i \in \mathbb{Z}$, and securely sends s_i to the player P_i .

2. *Distribution* D randomly selects two integer numbers c_1 and r , where $r \neq SC^j$ for $1 \leq j \leq k$, and sets $SC^0 = r$ (to ease the following recursive definition). Next, for secrets SC^1, \dots, SC^k defines:

$$\begin{cases} u_{j,0} = SC^j, u_{j,1} = f(SC^{j-1}, s_1), \dots, u_{j,\ell-2} = f(SC^{j-1}, s_{\ell-2}), \\ \sum_{\lambda=1}^{\ell} \binom{\ell-1}{\lambda-1} (-1)^\lambda u_{j,\ell+\ell-\lambda} = c_j \pmod p \quad (\ell \geq 0), \end{cases} \tag{3}$$

and computes $r_{j,i} = u_{j,i} - f(SC^{j-1}, s_i)$, for $\ell - 1 \leq i \leq n$. Finally, D publishes all $r, r_{j,i}$ for $1 \leq j \leq k$ and $\ell - 1 \leq i \leq n$.

3. *Reconstruction* To recover SC^j , at least ℓ^j participants provide the shadows $f(SC^{j-1}, s_i)$ in the j th stage. Then, they can solve the Vandermond system or use Lagrange interpolation to recover the secret SC^j .

In spite of the merits of this algorithm, the following cases makes it inappropriate for some applications mentioned earlier in this paper:

1. The Access structure of each secret is not arbitrary, i.e. $\Gamma_i \subseteq \Gamma_{i+1}$.
2. The previous secret is needed to reconstruct the next secret.
3. There is no mechanism to prevent cheating.
4. Secrets are recovered according to players' demands, not based on need or the development and the setup of the system.

2.5 Timed-Release Secret Sharing

As “time” is an essential part of our lives, associating time with cryptographic protocols seems quite useful and reasonable. One of the main objective of a timed-release cryptographic protocol is to safely transmit information to the future. Timed-release capability is investigated not only in encryption, but also in key-agreements and authentication codes, with information-theoretic security.

Watanabe and Shikata [24] presented the first Timed-Release Secret Sharing (TR-SS) method. The sharing phase of a TR-SS scheme takes a single secret and provides participants with private shares. To recover a secret in a TR-SS scheme, in addition to the cooperation of a qualified subset of shares, a time-signal, which is generated and published at a pre-specified time, is required as well.

In [24], two extra entities are employed: a Trusted Authority (TA) and a Time Server (TS). The TA generates and distributes secret keys for the dealer and the TS, and the TS broadcasts time-signals at a specific number of times.

The following steps represent a formal construction of a (t, n) -TR-SS scheme:

1. *Initializing* Assume $\mathcal{T} = \{1, 2, \dots, \tau\} \subset \mathbb{F}_p \setminus \{0\}$. First, the TA chooses the numbers $r^{(i)}$ ($i = 1, 2, \dots, \tau$) randomly from \mathbb{F}_p . The TA then sends a secret key $sk = (r^{(1)}, r^{(2)}, \dots, r^{(\tau)})$ to the TS and D via a secure channel.
2. *Sharing* The dealer determines the time e at which the qualified participants have permission to reconstruct the secret $SC \in \mathbb{F}_p$. Then, D constructs the polynomial $f(x) = c^{(e)} + \sum_{i=1}^{t-1} a_i x^i$ over \mathbb{F}_p , where a_i 's are uniformly selected from \mathbb{F}_p and $c^{(e)} = SC + r^{(e)}$. Eventually, D computes $SH_i^{(e)} = f(P_i)$ for $i = 1, 2, \dots, n$ and sends $SH_i = (SH_i^{(e)}, e)$ to the i th player, P_i , via a secure channel.

3. *Extracting* Based on sk and $d \in \mathcal{T}$, TS broadcasts the d th time signal $r^{(d)}$ at time d to all participants.
4. *Recovering* Any subset of at least t players can reconstruct $c^{(e)}$ using Lagrange interpolation. Finally, when the determined time for recovery comes, i.e. the appropriate time-signal comes, the secret is calculated by $SC = c^{(e)} - r^{(e)}$.

The above-mentioned TR-SS scheme is a single scheme and is strictly related to a pre-specified, fixed “time” slots.

3 The Proposed Algorithm

In this section, the details and phases of our algorithm are described. The proposed GSS algorithm contains two parts: the *sharing* phase and the *reconstruction* phase. In the sharing phase, secrets are shared and distributed among participants and a *trusted third party (TTP)* administrator. In the recovery phase, the first secret is recovered and if the TTP agrees with the process done by the first secret, shares his/her vote. Using this vote, participants can compute their shares for the second secret and the process repeats until all of the secrets are revealed or the process is terminated by the TTP. The following subsections will explain the details of the phases.

3.1 The Sharing Algorithm

Suppose that the reconstruction order is SC^1, SC^2, \dots, SC^k . The proposed framework does not impose any restrictions on the sub-parts of the scheme. However, without loss of generality, we make the following assumptions:

1. Each secret or each subset of secrets can have its own TTP, but we consider only one TTP for all secrets.
2. Secrets can be of different types, e.g. integers, audio files, images, etc. We restricted our analysis to integers.
3. Each step can recover more than one secret simultaneously; we recover only one secret at each step.

To apply the desired reconstruction order, D starts from the last secret, SC^k , makes a sharing scheme according to its access structure, Γ_k , then binds the shares using CRT. The outcome of the CRT and the TTP’s vote are used in a function to produce a temporary value. This temporary value along with the next to last secret, SC^{k-1} , are used to construct the next sharing structure. This process is repeated until all of the secrets are shared.

Figure 1 shows the flowchart of the proposed sharing phase. A useful characteristic of our new algorithm is that it does not need the scheme of all the secrets to be homogeneous. Every secret can be shared based on its own sharing scheme, its own access structure, and its own subset of participants. For integer type secrets, Shamir’s scheme [21] is preferred, properties of cellular automata makes them appropriate for sharing images and 3D objects [5, 8], and for computer graphics and geometric-aided designs Blackely’s geometry based sharing approach [2] is considered suitable [7]. Therefore, in the description we use $SA_{\Gamma_i}(\cdot)$ to represent the Sharing Algorithm, and $RA_{\Gamma_i}(\cdot)$ to represent the Reconstruction Algorithm corresponding to secret SC^i .

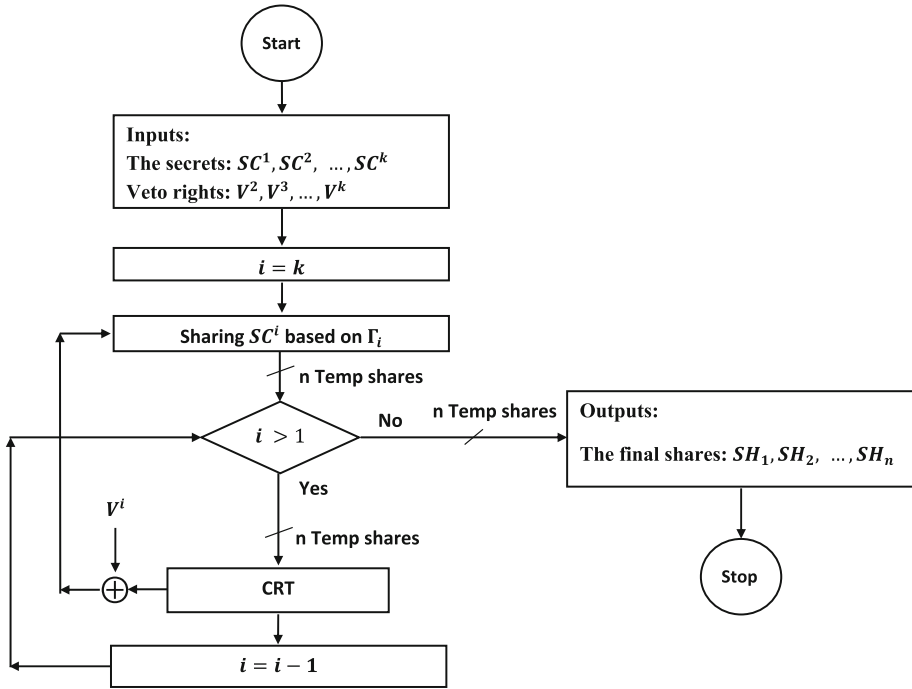


Fig. 1 The proposed sharing flowchart

The Dealer performs the following steps:

1. Chooses pairwise co-prime numbers m_1, m_2, \dots, m_n as moduli of the CRT.
2. Randomly selects votes V^2, V^3, \dots, V^k corresponding to secrets SC^2, SC^3, \dots, SC^k . Note that we assume that the first secret is SC^1 , and is assumed to be reachable without the permission of the TTP. If not, D simply chooses k values instead of $k - 1$ values.
3. Sets $i = k$ and T^i as a random value.
4. While $i \geq 1$ does the following steps:
 - (a) Constructs $SA_{\Gamma_i}(SC^i, T^i)$.
 - (b) Calculates the shares $SH_1^i, SH_2^i, \dots, SH_{n_i}^i$.
 - (c) Computes $C^i = CRT(SH_1^i, SH_2^i, \dots, SH_{n_i}^i)$.
 - (d) Sets $T^{i-1} = C^i \oplus V^i$, where “ \oplus ” is XOR operation.
 - (e) $i = i - 1$.
5. Sends V^2, V^3, \dots, V^k to the TTP and the shares $SH_1 = (SH_1^1, m_1), SH_2 = (SH_2^1, m_2), \dots, SH_n = (SH_n^1, m_n)$ with participants P_1, P_2, \dots, P_n via a secure channel.

3.2 The Reconstruction Algorithm

In the recovery phase, any qualified subset of participants listed in Γ_1 can reconstruct SC^1 using $RA_{\Gamma_1}(\cdot)$. They use this secret and progress somewhat through the first part of the work. If the TTP agrees with them in the first step, he/she publishes his/her veto right, V^2

for the second secret. Hereafter, qualified participants are able to obtain their shares by using their modulus, which is used to recover SC^2 . This process continues to recover all secrets. The gradual reconstruction can be blocked whenever the participants misuse the current secret, the TTP disagrees with the development of the system, or the shareholders are under attack by an adversary.

Our proposed recovery phase is preformed as follows:

1. Set $V^1 = all_zero_string$.
2. Set $i = 1$ and do the following steps while $i \leq k$:
 - (a) Given $Q \in \Gamma_i$, a qualified subset of eligible players for recovering secret SC^i . Compute SC^i using $(SC^i, T^i) = RA_{\Gamma_i}(Q)$.
 - (b) Using the recovered secret, SC^i , the first part of the process can be done.
 - (c) $i = i + 1$.
 - (d) If there is no problem with the operation and the TTP agrees to continue the process, she/he broadcasts V^i .
 - (e) Participants compute $C^i = V^i \oplus T_1^{i-1}$.
 - (f) Each participant P_j computes his/her share for the next secret as $SH_j^i = C^i \pmod{m_j}$
3. If $i > k$ then all secrets have been recovered, otherwise the TTP has blocked the process at step $b = i - 1$.

The function RA_{Γ_i} is the recovery algorithm corresponding to the SA_{Γ_i} . For example, if SA_{Γ_i} is Shamir's polynomial with a specific access structure, then RA_{Γ_i} is Lagrange interpolation with the same access structure. If one of the secrets used cellular automata as its SA_{Γ_i} , then the RA_{Γ_i} is the reverse cellular automata.

4 Security Analysis

The security of the proposed approach is based on Shamir's scheme as well as on the properties of the CRT. In this section, the security of our new algorithm is evaluated through analysis of different possible attacks against our scheme. We first list the main objectives of the proposed scheme and then we analyze to see if the scheme meets them. We expect that the new scheme:

1. Shares multiple secrets in an efficient and secure way;
2. Recovers secrets gradually in a pre-specified order, whenever they are needed;
3. Improves resistance against collusion of dishonest shareholders;
4. Recovers each secret based on its own access structure without any preconditions.

A number of properties of the proposed scheme depend on the SA_{Γ_i} and the RA_{Γ_i} . However, without loss of generality, we use Shamir's scheme in the following attacks analysis:

4.1 Attack

$t^i - 1$ or fewer shareholders want to recover the secret SC^i .

4.1.1 Analysis

Assume that t^i participants, for simplicity say $\mathcal{B} = \{P_1, P_2, \dots, P_{t^i}\}$ co-operate to recover SC^i . Each participant can calculate his/her share as $SH_j^i = C^i \pmod{m_j}$ for $j = 1, 2, \dots, t^i$. Thus, $SH_j^i = f^i(x_j)$, where $f^i(x) \in \mathbb{Z}_p[x]$ is the secret polynomial corresponding to secret SC^i . As $f^i(x)$ is a $t^i - 1$ degree polynomial, it can be written as Eq. 4, where coefficients $a_0^i, a_1^i, \dots, a_{t^i-1}^i$ are unknown elements of \mathbb{Z}_p and $a_0^i = SC^i$ is our intended secret.

$$f^i(x) = a_0^i + a_1^i x + \dots + a_{t^i-1}^i x^{t^i-1}. \tag{4}$$

Therefore, each participant in \mathcal{B} can obtain a linear equation with t^i unknowns $a_0^i, a_1^i, \dots, a_{t^i-1}^i$, i.e. they can form the following system of linear equations:

$$\begin{cases} a_0^i + a_1^i x_1 + \dots + a_{t^i-1}^i x_1^{t^i-1} = SH_1^i \\ a_0^i + a_1^i x_2 + \dots + a_{t^i-1}^i x_2^{t^i-1} = SH_2^i \\ \vdots \\ a_0^i + a_1^i x_{t^i-1} + \dots + a_{t^i-1}^i x_{t^i-1}^{t^i-1} = SH_{t^i-1}^i \end{cases} \tag{5}$$

In matrix form we have:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{t^i-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{t^i-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{t^i} & x_{t^i}^2 & \dots & x_{t^i}^{t^i-1} \end{pmatrix} \begin{pmatrix} a_0^i \\ a_1^i \\ \vdots \\ a_{t^i-1}^i \end{pmatrix} = \begin{pmatrix} SH_1^i \\ SH_2^i \\ \vdots \\ SH_{t^i}^i \end{pmatrix} \tag{6}$$

The coefficient matrix, A , is a Vandermonde matrix. There is a well-known formula for the determinant of a $t^i \times t^i$ Vandermonde matrix:

$$\det A = \prod_{1 \leq i < j \leq t^i} (x_i - x_j) \pmod{p}. \tag{7}$$

As it is assumed that the x_j s are distinct, $\det A \neq 0$, which implies that the system has a unique solution over \mathbb{Z}_p and any t^i participants can reconstruct polynomial $f^i(x)$ and obtain SC^i .

According to the above discussions, if $t^i - 1$ participants try to compute SC^i , then we have $t^i - 1$ linear equations with t^i unknowns. Suppose $SC^i = y_0$. Since secret $SC^i = a_0^i = f^i(0)$, we have

$$y_0 = f^i(0)$$

and this will be a (t^i)th linear equation. As before, there is now a unique solution to $f^i(x)$. Therefore, for every possible value y_0 of secret SC^i , there is a unique polynomial $f_{y_0}^i(x)$ such that

$$SH_j^i = f_{y_0}^i(x_j)$$

for $1 \leq j \leq t^i - 1$, and such that

$$y_0 = f_{y_0}^i(0).$$

In brief, no value of the secret can be ruled out by a group of $t^i - 1$ or fewer participants. \square

4.2 Attack

t^i colluding participants would try to obtain keys in not the prescribed arrangement.

4.2.1 Analysis

Assume $i - 1$ steps of the system have been executed and SC^1, \dots, SC^{i-1} were recovered in the proper order. Now players $\mathcal{B} = \{P_1, P_2, \dots, P_{t^i}\}$ try to discover another secret, other than SC^i . In such a situation, the only published information is the TTP's vote for SC^i , i.e. V^i . On the other hand, they obtained T^{i-1} by recovering the previous secret. According to the sharing algorithm (Sect. 3.1), they only can calculate $C^i = V^i \oplus T^i$. As C^i is computed, members of \mathcal{B} can only compute the shares related to SC^i . Calculating the shares corresponding to secrets SC^j , $i + 1 \leq j \leq k$ does not have probability better than $\frac{1}{M^{\frac{1}{M^j}}}$ where $M = \prod_{i=1}^n m_i$. Therefore, in a formal way,

$$H(SC^j | T^1, \dots, T^i, V^2, \dots, V^i) = H(SC^j), \tag{8}$$

where $H(\cdot)$ is the entropy function. \square

4.3 Attack

t^i colluding participants try to obtain the keys for all the stages of the system.

4.3.1 Analysis

To the best of our knowledge, this is the first time a solution to the problem of majority dishonest parties is being investigated. In the existing algorithms, t participants can learn all the secrets, i.e. all parts of the system. But in our new scheme, we can reduce the harm to just one secret without any extra cost, such as computing a one-way function, computing exponential operations, or even publishing further information. t^i dishonest participants just can obtain and misuse secret SC^i , as soon as the TTP notices this abuse, he/she can block the progression of the system. \square

4.4 Attack

Participants co-operated in reconstruction of SC^i try to obtain previous secrets: $SC^{i-1}, SC^{i-2}, \dots, SC^1$.

4.4.1 Analysis

Again, we assume that parties $\mathcal{B} = \{P_1, P_2, \dots, P_{t^i}\}$ are pooling their shares to calculate SC^i . As [18] acknowledged the access structures must be $\Gamma_i \subseteq \Gamma_{i+1}$, which means that all participants in the recovery of SC^i should contribute to recovering all, subsequent secrets. This might be undesirable in some applications.

Some of multi-stage schemes use the previous secret as a ticket to recovering the next secret [15, 18], this guarantees that recovery proceeds in the pre-specified order.

However, to prevent illegal access to the secrets our scheme has two separated parts: (1) the CRT part, in which the dealer specifies $n^i \subseteq n$, namely players who are authorize to recover SC^i , (2) the sharing part, which applies the access structure of each secret $SA_{\Gamma_i}/RA_{\Gamma_i}$ independent of the other access structures. \square

5 Comparisons

The proposed scheme offers some capabilities for the first time, hence the schemes being compared are not completely identical. Therefore, we acknowledge that this is not a completely fair comparison, but we try to emphasis on strengths of each scheme.

In SS schemes, it is quite popular to publish a number of values on the bulletin board. These public values are used later in the recovery phase, and the fewer the public values, the more efficient the scheme. In the most of the previous schemes [11, 15, 18, 22] these values are published by the dealer and are needed to be on bulletins for the full lifetime of the scheme. In scheme [24] and our proposed scheme, broadcasting information is done during the recovery phase and the newly published value replaces the previous one. Therefore, our scheme needs the public storage for just one value, i.e. for the current veto right.

Usually two factors are considered for efficiency analysis and comparison of secret sharing schemes: (1) *computational cost*, and (2) *communication cost*.

It is accepted practice to consider only heavy calculations and omit light operations in computational analysis. Accordingly, modular remaindering, adding or subtracting are usually ignored in the presence exponentiation or multiplication. The majority of computations in the proposed recovery are bitwise XOR, modular remaindering, and RA_{Γ_i} , which can be executed in efficient ways. As we based our reconstruction control on the CRT, the proposed scheme doesn't need costly computations in both sharing and recovering phases (Table 1).

Communication cost includes the number of bits that are transmitted during sharing and reconstruction phases. In the sharing phase, the communication cost includes the amount of information that is sent to the players as their shares plus the information that is published on a bulletin. In the reconstruction phase, we consider the quantity of information that is transmitted to recover all the secrets (gradually or simultaneously).

For the purpose of the illustration and ease of comparison, we have assumed that the hash function, one-way function, and the CRT module (M) are 1024-bit long and the shared secrets are 160-bit long. These comparisons shows the efficiency of our proposed scheme, where computational cost of sharing and recovering are almost negligible, and communication cost of sharing and recovering are of order $\mathcal{O}(n+k)$ and $\mathcal{O}(\sum_{i=1}^k t^i)$ respectively.

6 Conclusions

Multi-secret sharing schemes are a very useful concept for settings with multiple parts, requiring having different keys for each part. Traditional methods recover secrets in one step, increasing the chance of misuse. A new efficient multi-secret sharing platform based on CRT and a number of arbitrary single sharing schemes is presented in this paper. Our scheme provides a completely free-access structure for each secret, and has a strong

Table 1 Comparison of basic properties of investigated papers

Property	Mashhadi [18]	Shao [22]	TR-SS [24]	GSS
Many secrets are shared	✓	✓	-	✓
Gradual reconstruction	✓	-	-	✓
Controlling collusion of majority of players	-	-	-	✓
Number of leaked keys in collusion of at least t players	k	k	-	1
Arbitrary access structure for each secret	-	-	-	✓
Required number of public storages	kn	$n + 3k - t$	1	1
Computational cost (sharing)	kn	$n + 1$	0	0
Computational cost (recovering)	$\sum_{i=1}^k (t^i)$	t	0	0
Communication cost (sharing)	$160(n+1) + (k(n+2) - \sum_{i=1}^k t^i)1024$ $\simeq \mathcal{O}(kn)$	$320n + 1024(n+t)$ $\simeq \mathcal{O}(n+t)$	$160(2\tau^{(*)} + n)$ $\simeq \mathcal{O}(\tau+n)$	$160n + 1024(k-1)$ $\simeq \mathcal{O}(n+k)$
Communication cost (recovering)	$(\sum_{i=1}^k t^i)1024$ $\simeq \mathcal{O}(\sum_{i=1}^k t^i)$	$320t$ $\simeq \mathcal{O}(t)$	$160(t + e^{(*)})$ $\simeq \mathcal{O}(t+e)$	$160(\sum_{i=1}^k t^i + 1024(k-1))$ $\simeq \mathcal{O}(\sum_{i=1}^k t^i)$

* In TR-SS [24], the third party publishes time-signals until the desired time comes. Therefore, τ and e are much larger than parameters like k , n , or t

resistance to collusions among dishonest participants. It is rather efficient, as it requires a small and fixed amount of public storage and very minimal computational overhead. The new scheme is easily constructed and it is suitable for situations where players have limited amenities. Its two stage approach to recovering attack can decrease possibility of information leakage.

References

1. Asmuth, C., & Bloom, J. (1983). A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, 30(2), 208–210.
2. Blakely, G. (1979). Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS national computer conference* (pp. 313–317).
3. Cohen, H. (2013). *A course in computational algebraic number theory* (4th ed., Vol. 138). Berlin: Springer.
4. Dehkordi, M. H., & Ghasemi, R. (2016). A lightweight public verifiable multi secret sharing scheme using short integer solution. *Wireless Personal Communications*, 91(3), 1459–1469.
5. del Rey, A. M. (2015). A multi-secret sharing scheme for 3D solid objects. *Expert Systems with Applications*, 42(4), 2114–2120.
6. Drăgan, C. C., & Tiplea, F. L. (2016). Distributive weighted threshold secret sharing schemes. *Information Sciences*, 339, 85–97.
7. Elsheh, E., & Hamza, A. B. (2011). Secret sharing approaches for 3D object encryption. *Expert Systems with Applications*, 38(11), 13906–13911.
8. Eslami, Z., & Ahmadabadi, J. Z. (2010). A verifiable multi-secret sharing scheme based on cellular automata. *Information Sciences*, 180(15), 2889–2894.
9. Eslami, Z., & Ahmadabadi, J. Z. (2011). Secret image sharing with authentication-chaining and dynamic embedding. *Journal of Systems and Software*, 84(5), 803–809.
10. Eslami, Z., Razzaghi, S., & Ahmadabadi, J. Z. (2010). Secret image sharing based on cellular automata and steganography. *Pattern Recognition*, 43(1), 397–404.
11. Fatemi, M., Ghasemi, R., Eghlidos, T., & Aref, M. R. (2014). Efficient multistage secret sharing scheme using bilinear map. *IET Information Security*, 8(4), 224–229.
12. Harn, L., Fuyou, M., & Chang, C.-C. (2014). Verifiable secret sharing based on the Chinese remainder theorem. *Security and Communication Networks*, 7(6), 950–957.
13. Harn, L., & Hsu, C.-F. (2017). (t, n) multi-secret sharing scheme based on bivariate polynomial. *Wireless Personal Communications*, 95(2), 1495–1504.
14. Harn, L., Lin, C., & Li, Y. (2015). Fair secret reconstruction in (t, n) secret sharing. *Journal of Information Security and Applications*, 23, 1–7.
15. He, J., & Dawson, E. (1994). Multistage secret sharing based on one-way function. *Electronics Letters*, 30(19), 1591–1592.
16. Hsu, C.-F., Harn, L., & Cui, G. (2014). An ideal multi-secret sharing scheme based on connectivity of graphs. *Wireless Personal Communications*, 77(1), 383–394.
17. Hua, W., & Liao, X. (2017). A secret image sharing scheme based on piecewise linear chaotic map and Chinese remainder theorem. *Multimedia Tools and Applications*, 76(5), 7087–7103.
18. Mashhadi, S. (2016). How to fairly share multiple secrets stage by stage. *Wireless Personal Communications*, 90(1), 93–107.
19. Mashhadi, S., & Dehkordi, M. H. (2015). Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem. *Information Sciences*, 294, 31–40.
20. Pilaram, H., & Eghlidos, T. (2017). An efficient lattice based multi-stage secret sharing scheme. *IEEE Transactions on Dependable and Secure Computing*, 14(1), 2–8.
21. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11), 612–613.
22. Shao, J. (2014). Efficient verifiable multi-secret sharing scheme based on hash function. *Information Sciences*, 278, 104–109.
23. Tian, Y., Ma, J., Peng, C., & Jiang, Q. (2013). Fair (t, n) threshold secret sharing scheme. *IET Information Security*, 7(2), 106–112.
24. Watanabe, Y., & Shikata, J. (2016). Information-theoretically secure timed-release secret sharing schemes. *Journal of Information Processing*, 24(4), 680–689.

25. Zarepour-Ahmadabadi, J., Ahmadabadi, M. S., & Latif, A. (2016). An adaptive secret image sharing with a new bitwise steganographic property. *Information Sciences*, 369, 467–480.



Jamal Zarepour-Ahmadabadi received his B.S. degree in Management Information System (MIS) in 2006 from Yazd University. He graduated as M.S. of Computer Science from Shahid Beheshti University, Tehran, Iran. Now, he is a Ph.D. candidate of Computer Science in Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. His research interests are Cryptography, Information Security and Artificial Intelligence.



MohammadEbrahim Shiri-Ahmadabadi is an Assistant Professor with Department of Computer Science of Amirkabir University of Technology. His research interests are Artificial Intelligent, Image Processing, Vision, Machine Learning and their applications.



Ali Miri has been a Full Professor at the School of Computer Science, Ryerson University, Toronto since 2009. He is the Research Director, Privacy and Big Data Institute, Ryerson University, an Affiliated Scientist at Li Ka Shing Knowledge Institute, St. Michaels Hospital, and a member of Standards Council of Canada, Big Data Working Group. He has also been with the School of Information Technology and Engineering and the Department of Mathematics and Statistics since 2001, and has held visiting positions at the Fields Institute for Research in Mathematical Sciences, Toronto in 2006, and Universite de Cergy-Pontoise, France in 2007, and Alicante and Albcete Universities in Spain in 2008. His research interests include cloud computing and big data, computer networks, digital communication, and security and privacy technologies and their applications. He has authored and co-authored more than 180 referred articles, 6 books, and 5 patents in these fields. Dr. Miri has chaired over a dozen international conference and workshops, and had served on more than 80 technical program committees. He is a senior member of the IEEE, and a member of the Professional Engineers Ontario.



Ali Mohammad Latif obtained his first degree in Electronic Engineering from Isfahan University of Technology in 1993 and his M.Sc. in Electronic Engineering from Amirkabir University of Technology in 1996. In 2001, he joined the academic staff at electrical engineering department of Yazd University. He worked as a Ph.D. candidate at the University of Isfahan where he obtained his Ph.D. In 2011, he joined the academic staff at computer department of Yazd University. His research interests are Digital Image Processing, watermarking and Cryptography.