JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 749-763 (2018)
DOI: 10.6688/JISE.201805_34(3).0011

On the Construction and Analysis of Verifiable Multi-secret Sharing Based on Non-homogeneous Linear Recursion*

BEN-HUI ZHANG[1] AND YUAN-SHENG TANG[2]
[1]*School of Mathematical Science*
Huaibei Normal University
Huaibei, P.R. 235000 China
[2]*School of Mathematical Science*
Yangzhou University
Yangzhou, P.R. 225002 China
E-mail: benhuizhang@163.com; ystang@yzu.edu.cn

We shall propose two verifiable multi-secret sharing schemes based on novel non-homogeneous linear recursions. In the initial phase, the secret shadow of each participant is selected by himself. In the construction phase, the dealer puts the information of shared secrets in the initialization vector of non-homogeneous linear recursions of degree $t-1$. In the verification phase, verification algorithms based on elliptic curve are designed to resist a variety of cheating actions or attacks. In the recovery phase, each participant just needs to provide secret share instead of secret shadow. The proposed schemes have the following features: verifiability, the reuse of secret shadows and shared secrets is possible, only public channels are needed. Compared with previous schemes, they have better performance, fewer public values, lower computation complexity, shorter key length and less running time.

Keywords: secret sharing, non-homogeneous linear recursion, public value, computation complexity, running time

1. INTRODUCTION

In 1979, Blakley [1] and Shamir [2] gave the construction for secret sharing based on linear projective geometry and interpolation polynomial, respectively. A secret sharing scheme is a mechanism for sharing a secret among a group of participants so that only participants in an authorized subset can recover the secret whereas participants in an unauthorized subset cannot obtain anything about the secret. In a (t, n) threshold secret sharing scheme, t or more participants can cooperate to recover the secret, but $t-1$ or fewer participants cannot. In a multi-secret sharing (MSS) scheme, multiple secrets can be shared during one sharing process. Chien [3] proposed a MSS scheme based on the systematic block codes. Yang [4] proposed an alternative (t, n) MSS scheme based on Shamir secret sharing. If $k > t$, the dealer needs to construct a polynomial of degree $k-1$ and publish $n+k-t+1$ values; if $k \le t$, the dealer needs to construct a polynomial of degree $t-1$ and publish $n+1$ values. Shao and Cao [5] pointed the security problem of scheme [4] and proposed a verifiable multi-secret sharing (VMSS) scheme. Zhao [6] and Dehkordi [7] not only solved the security problem, but also used public channels instead

Received October 24, 2016; revised January 8 & March 23, 2017; accepted April 28, 2017.
Communicated by Hung-Min Sun.

* This work was supported by the National Natural Science Foundation of China (61379004), Natural Science Foundation of Anhui Higher Education Institutions of China (KJ2016A634).

of secret channels to transmit information. The running time of recovery algorithms in [4-7] is $\mathcal{O}(k^2)(k > t)$ or $\mathcal{O}(t^2)(k \leq t)$. In 2008, Dehkordi [8] proposed two computationally secure (t, n) VMSS schemes based on homogeneous linear recursion (HLR). The degree of reconstruction polynomial of these two schemes is both $t - 1$, and the number of public values is $2n + k - t + 1$ and $2n + k - t + 2$, respectively. Chen [9] improved the schemes [8] by reducing the number of public values to $n + k + 1$ and Hu [10] also improved the schemes [8] by decreasing the key length. In 2016, Liu [11] showed that schemes [6-8] cannot withstand some deceptive behaviors of the dealer and gave two improved schemes based on RSA public key cryptosystem. In 2008, Dehkordi [12] proposed two (t, n) VMSS schemes based on non-homogeneous linear recursion (NHLR) and ECRSA cryptosystem. In 2015, Mashhadi [13] gave new constructions for secret sharing based on NHLR and LFSR public key cryptosystem, which improved the schemes [12] by decreasing the number of public values from $2n + k - t + 4$ to $2n + k - t + 3$ and the degree of reconstruction polynomial from $t + 1$ to $t - 1$. Although verifiable secret sharing (VSS) scheme can be used to detect the cheating, it cannot prevent the vicious participant from obtaining the true secret. It is unfair for other participants who provide correct shares since these participants just obtain a wrong secret. In 1988, Tompa and Woll [14] proposed the first fair secret sharing (FSS) scheme, in which the real secret s is hidden in a sequence of identical dummy secret. The major concern of this scheme is that all participants must simultaneously release their shares. In other words, the cheater may get the secret exclusively if he releases his share last in an asynchronous model. In 2013, Tian [15] designed a fair threshold scheme. They used the similar approach of Tompa and Woll [14] to achieve the fairness and utilize the redundancy of shares to detect the cheating. They proved its fairness against non-cooperative attack, cooperative attack with synchronization and cooperative attack with asynchronization. Harn [16] pointed out that Tian's scheme [15] just works properly in a synchronous model, but not in an asynchronous model. Harn [17] constructed an asynchronous threshold scheme against outside attacker and extended it to multi-secret sharing. There is no cryptographic assumption in [17], so it is unconditionally secure. In 2015, Harn [18] designed an asynchronously rational secret sharing scheme against both inside and outside attackers. They also followed the approach of scheme [17] to distribute shares and utilized the collision-resistant one-way function to detect cheating.

In this paper, we use the new non-homogeneous linear recursions of degree $t - 1$ to construct two multi-secret sharing schemes. We also design new verification algorithms based on elliptic curve to detect the cheating actions or attacks. In the proposed schemes, the number of public values is $n + k + 2$, the degree of reconstruction polynomial is $t - 1$ and the running time of recovery phase is $\mathcal{O}(t^2)$. Compared with existing schemes [8-13] based on linear recursion, the proposed schemes have fewer public values, lower computation complexity and shorter key length. Compared with schemes [4-7, 10, 11] based on Lagrange interpolation polynomial, our proposed schemes have better performance and less running time.

The rest of this paper is organized as follows. Section 2 introduces related concepts and results of elliptic curves and non-homogeneous linear recursion. Section 3 describes the proposed multi-secret sharing schemes. Section 4 gives the security and performance analysis of the proposed schemes. Section 5 discusses the cryptographic properties with other related schemes. Section 6 concludes this paper.

2. PRELIMINARIES

2.1 Elliptic Curves

In this section, we revisit some concepts about elliptic curves. A detailed description can be found in [19].

Definition 1: An elliptic curve $E_p(a, b)$ over the finite field $\mathbb{Z}_p$ is given through an equation: $y^2 \equiv x^3 + ax + b(\text{mod}\, p)$, where $a, b \in \mathbb{Z}_p$ satisfying $\gcd(4a^3 + 27b^2, p) = 1$, that is

$$E_p(a, b) = \{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p\text{: } y^2 \equiv x^3 + ax + b(\text{mod}\, p)\} \cup \{\mathrm{O}\}$$

here O is the point at infinity.

The points of an elliptic curve $E_p(a, b)$ make up a group under the algebraic structure defined as follows:

(1) Identity element: define the point at infinity O as the identity element, *i.e.*, $\forall P \in E_p(a, b)$, $P + \mathrm{O} = \mathrm{O} + P = P$.

(2) Inverse element: if $P = (x, y) \neq \mathrm{O}$; $-P = (x, -y) \neq \mathrm{O}$; $-\mathrm{O} = \mathrm{O}$.

(3) Addition operation: let

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2),\ P_1 \neq -P_2, \text{ then } P_1 + P_2 = (x_3, y_3) \in E_p(a, b),$$

where

$$\begin{cases} x_3 = \left(\lambda^2 - x_1 - x_2\right) \text{mod}\, p \\ y_3 = \left(\lambda\left(x_1 - x_3\right) - y_1\right) \text{mod}\, p \end{cases}, \text{ here } \lambda = \begin{cases} \left(y_2 - y_1\right)\left(x_2 - x_1\right)^{-1} \text{mod}\, p, & x_1 \neq x_2 \\ \left(3x_1^2 + a\right)\left(2y_1\right)^{-1} \text{mod}\, p, & x_1 = x_2 \end{cases}.$$

Definition 2: The order of a point $Q \in E_p(a, b)$ is defined as the minimum integer l that satisfies $lQ = \mathrm{O}$, here $lQ = \underbrace{Q + \cdots + Q}_{l}\cdot$

Definition 3: Suppose the order of $Q \in E_p(a, b)$ is l (l is a large prime number), for a given point $R \in \langle Q \rangle$, to calculate the integer $n \in [1, l]$ satisfying $nQ = R$ is called elliptic curve discrete logarithm problem (ECDLP).

2.2 Non-Homogeneous Linear Recursion

In this section, we revisit the concept and some results about non-homogeneous linear recursion. A detailed description can be found in [13, 20].

Definition 4: Let $c_0, c_1, \ldots, c_{t-1}, a_1, a_2, \ldots, a_t$ be real numbers and t be a positive integer, a non-homogeneous linear recursion (NHLR) of degree t is defined by the following equations:

$$\begin{cases} u_0 = c_0, u_1 = c_1, \cdots, u_{t-1} = c_{t-1}, \\ u_{i+t} + a_1 u_{i+t-1} + \cdots + a_t u_i = f\left(i\right)\left(i \geq 0\right), \end{cases}$$

here $c_0, c_1, \ldots, c_{t-1}, a_1, a_2, \ldots, a_t$ are constants.

The terms of the sequence defined by NHLR depend on the initialization vector $(u_0, u_1, \ldots, u_{t-1})$ and the coefficients $a_i (1 \leq i \leq t)$ as well as the function $f(i)$.

Lemma 1: Let $\mathbb{F}$ be a field, $h(x) \in \mathbb{F}(x)$ is a polynomial. Consider a typical fraction $h(x)/(1 - \alpha x)^m$, $\alpha \in \mathbb{F}$, $\deg(h(x)) < m$, $q > m$, here q is the characteristic of the field $\mathbb{F}$. Then $\frac{h(x)}{(1-\alpha x)^m} = \sum_{i=0}^{\infty} u_i x^i$, where $u_i = p(i)\alpha^i$ and $p(i) = A_0 + A_1 i + \ldots + A_{m-1} i^{m-1}$.

Theorem 1: Suppose the sequence (u_i) is defined by the following NHLR:

$$\begin{cases} u_0 = c_0, u_1 = c_1, \cdots, u_{t-2} = c_{t-2}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} u_{i+t-j-1} = (-1)^i c (i \geq 0), \end{cases}$$

where $c_0, c_1, \ldots, c_{t-2}, c$ are constant real numbers. Then $u_i = (-1)^i p(i)$, here $p(i) = A_0 + A_1 i + \ldots + A_{t-1} i^{t-1}$.

Theorem 2: Suppose the sequence (u_i) is defined by the following NHLR:

$$\begin{cases} u_0 = c_0, u_1 = c_1, \cdots, u_{t-2} = c_{t-2}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} (-1)^j u_{i+t-j-1} = c (i \geq 0), \end{cases}$$

where $c_0, c_1, \ldots, c_{t-2}, c$ are constant real numbers. Then $u_i = p(i)$, here $p(i) = A_0 + A_1 i + \ldots + A_{t-1} i^{t-1}$.

3. NEW VMSS SCHEMES

In this section, we design two verifiable (t, n) threshold multi-secret sharing schemes based on new non-homogeneous linear recursions, which can be viewed as variants and improvements of Dehkordi-Mashhadi [12] and Mashhadi-Dehkordi [13].

3.1 Type 1 Scheme

We use the non-homogeneous linear recursion of Theorem 1 to construct Type 1 scheme.

Initialization

$P_0, P_1, \ldots, P_{k-1}$ denote k secrets and $M_1, M_2, \ldots, M_n$ denote n participants. The dealer D selects a large prime number p based on the security of elliptic curve cryptography. $E_p(a, b)$ is an elliptic curve defined over the prime field $\mathbb{Z}_p$. D also chooses a point $Q \in E_p(a, b)$ of order q such that the elliptic curve discrete logarithm problem is intractability in $\langle Q \rangle$, here $q (> t)$ is a large prime number. The dealer D publishes $\{p, Q\}$.

Every participant M_i randomly chooses an integer s_i as his own secret shadow and computes $R_i = s_iQ$, then transmits (R_i, i) to the dealer D. D must ensure that $R_i \neq R_j$ for all $i \neq j$, then randomly selects an integer d such that $R_0 \neq R_i (i = 1, 2, \ldots, n)$, here $R_0 = dQ$. D also computes $B_i = dR_i = (x_{B_i}, y_{B_i})$ and $I_i = x_{B_i} + y_{B_i}$ for $i = 1, 2, \ldots, n$.

Construction Phase

If $k \leq t - 1$, the dealer D executes the following steps:

(1) Consider the following non-homogeneous linear recursion of degree $t - 1$:

$$\begin{cases} u_0 = P_0, \cdots, u_{k-1} = P_{k-1}, u_k = I_1, \cdots, u_{t-2} = I_{t-k-1}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} u_{i+t-j-1} = (-1)^i c \bmod q, i \geq 0, c \in \mathbb{Z}. \end{cases}$$

(2) Compute u_i for $t - 1 \leq i \leq n + k - 1$.
(3) Compute $y_i = I_i - u_{k+i-1}$ for $t - k \leq i \leq n$.
(4) Use t pairs $\{X_i = i - 1, Y_i = (-1)^{i-1}u_{i-1}\}_{i=1}^t$ to construct a polynomial $p(x)$ of degree $t - 1$:

$$p(x) = \sum_{i=1}^{t} Y_i \prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1x^1 + \cdots + A_{t-1}x^{t-1} \bmod q.$$

Compute $C_j = A_jQ$ for $0 \leq j \leq t-1$.
(5) Publish $\{R_0, y_{t-k}, \ldots, y_n, C_0, \ldots, C_{t-1}\}$.

If $k > t - 1$, the dealer D executes the following steps:

(1) Consider the following non-homogeneous linear recursion of degree $t - 1$:

$$\begin{cases} u_0 = P_0, u_1 = P_1, \cdots, u_{t-2} = P_{t-2}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} u_{i+t-j-1} = (-1)^i c \bmod q, i \geq 0, c \in \mathbb{Z}. \end{cases}$$

(2) Compute u_i for $t - 1 \leq i \leq n + k - 1$.
(3) Compute $r_i = P_i - u_i$ for $t - 1 \leq i \leq k - 1$, $y_i = I_i - u_{k+i-1}$ for $1 \leq i \leq n$.
(4) Use t pairs $\{X_i = i - 1, Y_i = (-1)^{i-1}u_{i-1}\}_{i=1}^t$ to construct a polynomial $p(x)$ of degree $t - 1$:

$$p(x) = \sum_{i=1}^{t} Y_i \prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1x^1 + \cdots + A_{t-1}x^{t-1} \bmod q.$$

Compute $C_j = A_jQ$ for $0 \leq j \leq t - 1$.
(5) Publish $\{R_0, y_1, \ldots, y_n, r_{t-1}, \ldots, r_{k-1}, C_0, \ldots, C_{t-1}\}$.

Verification Phase

Each participant M_i can compute s_iR_0 to obtain his secret share B_i. Suppose t participants $\{M_i\}_{i\in I}(I \subseteq \{1, \ldots, n\}, |I| = t)$ pool their shares $\{B_i\}_{i\in I}$ to reconstruct the secrets $P_0, P_1, \ldots, P_{k-1}$.

If $k \le t-1$, each participant $M_i(i\in I)$ executes the following steps to verify the secret share B_j for $j\neq i, j\in I$:

(1) Compute $I_j = x_{B_j} + y_{B_j}$.
(2) Compute $u_{k+j-1} = \begin{cases} I_j, & 1 \le j \le t-k-1 \\ I_j - y_j, & t-k \le j \le n \end{cases}$.
(3) Check the secret share by the following equation:

$$u_{k+j-1}Q = (-1)^{k+j-1} \sum_{l=0}^{t-1} (k+j-1)^l C_l, j \ne i, j \in I .$$

If $k > t-1$, each participant $M_i(i\in I)$ executes the following steps to verify the secret share B_j for $j\neq i, j\in I$:

(1) Compute $I_j = x_{B_j} + y_{B_j}$.
(2) Compute $u_{k+j-1} = I_j - y_j$.
(3) Check the secret share by the following equation:

$$u_{k+j-1}Q = (-1)^{k+j-1} \sum_{l=0}^{t-1} (k+j-1)^l C_l, j \ne i, j \in I.$$

Recovery Phase

Here we present two methods to recover the shared secrets.

Method 1: Using Lagrange interpolation polynomial.

Suppose t arbitrary participants $\{M_i\}_{i\in I}(I\subseteq\{1, \ldots, n\}, |I| = t)$ pool their shares $\{B_i\}_{i\in I}$ and all the shares pass the verification phase. They can get t terms $\{u_{k+i-1}\}_{i\in I}$, then they use t pairs $\{X_i = k+i-1, Y_i = (-1)^{k+i-1}u_{k+i-1}\}_{i\in I}$ to construct the following polynomial $p(x)$ of degree $t-1$ and compute $u_j = (-1)^j p(j) \bmod q$ for $j = 0, 1, \ldots, k-1$.

$$p(x) = \sum_{i\in I} Y_i \prod_{j\in I, j\ne i} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1 x^1 + \cdots + A_{t-1}x^{t-1} \bmod q.$$

If $k \le t-1$, they recover the shared secrets as $P_j = u_j, j = 0, 1, \ldots., k-1$.

If $k > t-1$, they recover the shared secrets as $P_j = \begin{cases} u_j, & j = 0,1,\cdots,t-2 \\ r_j + u_j, & j = t-1, t, \cdots, k-1 \end{cases}$.

Method 2: Using NHLR.

Suppose t successive participants $\{M_i, M_{i+1}, \ldots, M_{i+t-1}\}$, $(1 \le i \le n-t+1)$ pool their shares $\{B_i, B_{i+1}, \ldots, B_{i+t-1}\}$ and all the shares pass the verification phase. They can get t terms $\{u_{k+i-1}, u_{k+i}, \ldots, u_{k+i+t-2}\}$ and consider the NHLR (Δ):

$$\sum_{j=0}^{t-1} \binom{t-1}{j} u_{l+t-j-1} = (-1)^l c \bmod q . \qquad (\Delta)$$

In NHLR(Δ), they can use $u_{k+i-1}, u_{k+i}, \ldots, u_{k+i+t-2}$ to compute c by taking $l = k + i - 1$, then they compute $u_{k+i-2}, \ldots, u_{k-1}, \ldots, u_1, u_0$ through the following equations:

$$u_l = (-1)^l c - \sum_{j=0}^{t-2} \binom{t-1}{j} u_{l+t-j-1} \bmod q,\ l = k+i-2,\ldots,k-1,\ldots,1,0.$$

If $k \le t - 1$, the shared secrets can be recovered as $P_j = u_j$, $j = 0, 1, \ldots, k - 1$.

If $k > t - 1$, the shared secrets can be recovered as $P_j = \begin{cases} u_j, & j = 0,1,\cdots,t-2 \\ r_j + u_j, & j = t-1,t,\cdots,k-1 \end{cases}.$

3.2 Type 2 Scheme

We use the non-homogeneous linear recursion of Theorem 2 to construct Type 2 scheme.

Initialization

The initialization in this type is the same as that in previous type.

Construction Phase

If $k \le t - 1$, the dealer D executes the following steps:

(1) Consider the following non-homogeneous linear recursion of degree $t - 1$:

$$\begin{cases} u_0 = P_0,\cdots,u_{k-1} = P_{k-1}, u_k = I_1,\cdots,u_{t-2} = I_{t-k-1}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} (-1)^j u_{i+t-j-1} = c \bmod q, i \ge 0, c \in \mathbb{Z}. \end{cases}$$

(2) Compute u_i for $t - 1 \le i \le n + k - 1$.
(3) Compute $y_i = I_i - u_{k+i-1}$ for $t - k \le i \le n$.
(4) Use t pairs $\{X_i = i - 1, Y_i = u_{i-1}\}_{i=1}^{t}$ to construct a polynomial $p(x)$ of degree $t - 1$:

$$p(x) = \sum_{i=1}^{t} Y_i \prod_{j=1,j\ne i}^{t} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1 x^1 + \cdots + A_{t-1} x^{t-1} \bmod q.$$

Compute $C_j = A_j Q$ for $0 \le j \le t - 1$.
(5) Publish $\{R_0, y_{t-k}, \ldots, y_n, C_0, \ldots, C_{t-1}\}$.

If $k > t - 1$, the dealer D executes the following steps:

(1) Consider the following non-homogeneous linear recursion of degree $t - 1$:

$$\begin{cases} u_0 = P_0, u_1 = P_1,\cdots,u_{t-2} = P_{t-2}, \\ \sum_{j=0}^{t-1} \binom{t-1}{j} (-1)^j u_{i+t-j-1} = c \bmod q, i \ge 0, c \in \mathbb{Z}. \end{cases}$$

(2) Compute u_i for $t - 1 \le i \le n + k - 1$.

(3) Compute $r_i = P_i - u_i$ for $t-1 \le i \le k-1$, $y_i = I_i - u_{k+i-1}$ for $1 \le i \le n$.
(4) Use t pairs $\{X_i = i-1, Y_i = u_{i-1}\}_{i=1}^{t}$ to construct a polynomial $p(x)$ of degree $t-1$:

$$p(x) = \sum_{i=1}^{t} Y_i \prod_{j=1, j\ne i}^{t} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1 x^1 + \cdots + A_{t-1} x^{t-1} \bmod q.$$

Compute $C_j = A_j Q$ for $0 \le j \le t-1$.
(5) Publish $\{R_0, y_1, \ldots, y_n, r_{t-1}, \ldots, r_{k-1}, C_0, \ldots, C_{t-1}\}$.

Verification Phase

Each participant M_i can compute $s_i R_0$ to obtain his secret share B_i. Suppose t participants $\{M_i\}_{i\in I}(I \subseteq \{1, \ldots, n\}, |I| = t)$ pool their shares $\{B_i\}_{i\in I}$ to reconstruct the secrets $P_0, P_1, \ldots, P_{k-1}$.

If $k \le t-1$, each participant $M_i(i\in I)$ executes the following steps to verify the secret share B_j for $j\ne i, j\in I$:

(1) Compute $I_j = x_{B_j} + y_{B_j}$.
(2) Compute $u_{k+j-1} = \begin{cases} I_j, & 1 \le j \le t-k-1 \\ I_j - y_j, & t-k \le j \le n \end{cases}$.
(3) Check the secret share by the following equation:

$$u_{k+j-1} Q = \sum_{l=0}^{t-1} (k+j-1)^l C_l, j \ne i, j \in I.$$

If $k > t-1$, each participant $M_i(i\in I)$ executes the following steps to verify the secret share B_j for $j \ne i, j \in I$:

(1) Compute $I_j = x_{B_j} + y_{B_j}$.
(2) Compute $u_{k+j-1} = I_j - y_j$.
(3) Check the secret share by the following equation:

$$u_{k+j-1} Q = \sum_{l=0}^{t-1} (k+j-1)^l C_l, j \ne i, j \in I.$$

Recovery phase

Here we present two methods to reconstruct the shared secrets.

Method 1: Using Lagrange interpolation polynomial

Suppose t arbitrary participants $\{M_i\}_{i\in I}(I \subseteq \{1, \ldots, n\}, |I| = t)$ pool their shares $\{B_i\}_{i\in I}$ and all the shares pass the verification phase. They can get t terms $\{u_{k+i-1}\}_{i\in I}$, then they use t pairs $\{X_i = k+i-1, Y_i = u_{k+i-1}\}_{i\in I}$ to construct the following polynomial $p(x)$ of degree $t-1$ and compute $u_j = p(j)$ for $j = 0, 1, \ldots, k-1$.

$$p(x) = \sum_{i\in I} Y_i \prod_{j\in I, j\ne i} \frac{x - X_j}{X_i - X_j} \bmod q = A_0 + A_1 x^1 + \cdots + A_{t-1} x^{t-1} \bmod q.$$

If $k \le t - 1$, they recover the shared secrets as $P_j = u_j, j = 0, 1, \ldots, k - 1$.

If $k > t - 1$, they recover the shared secrets as $P_j = \begin{cases} u_j, & j = 0,1,\cdots,t-2 \\ r_j + u_j, & j = t-1,t,\cdots,k-1 \end{cases}$.

Method 2: Using NHLR.

Suppose t successive participants $\{M_i, M_{i+1}, \ldots, M_{i+t-1}\}$, $(1 \le i \le n - t + 1)$ pool their shares $\{B_i, B_{i+1}, \ldots, B_{i+t-1}\}$ and all the shares pass the verification phase. They can get t terms $\{u_{k+i-1}, u_{k+i}, \ldots, u_{k+i+t-2}\}$ and consider the NHLR (Ω):

$$\sum_{j=0}^{t-1}\binom{t-1}{j}(-1)^j u_{l+t-j-1} = c \bmod q. \qquad (\Omega)$$

In NHLR (Ω), they can use $u_{k+i-1}, u_{k+i}, \ldots, u_{k+i+t-2}$ to compute c by taking $l = k + i - 1$, then they compute $u_{k+i-2}, \ldots, u_{k-1}, \ldots, u_1, u_0$ through the following equations:

$$u_l = (-1)^{1+t} c + \sum_{j=0}^{t-2}\binom{t-1}{j}(-1)^{t+j} u_{l+t-j-1} \bmod q, l = k+i-2,\cdots,k-1,\cdots,1,0.$$

If $k \le t - 1$, they recover the shared secrets as $P_j = u_j, j = 0, 1, \ldots, k - 1$.

If $k > t - 1$, the shared secrets can be recovered as $P_j = \begin{cases} u_j, & j = 0,1,\cdots,t-2 \\ r_j + u_j, & j = t-1,t,\cdots,k-1 \end{cases}$.

4. SECURITY AND PERFORMANCE ANALYSIS

4.1 Security Analysis

Attack 1: $t - 1$ or fewer participants may try to recover the shared secrets.

Discussions: In our schemes, there are two methods to reconstruct the shard secrets. For method 1: using Lagrange interpolation polynomial, suppose $m(1 \le m \le t - 1)$ arbitrary participants $\{M_i\}_{i\in I}(|I| = m)$ pool their secret shares. They can obtain m pairs $\{k + i - 1, (-1)^{k+i-1}u_{k+i-1}\}_{i\in I}$ in Type 1 scheme and they can also obtain m pairs $\{k + i - 1, u_{k+i-1}\}_{i\in I}$ in type 2 scheme. That is to say, the number of obtained pairs is less than t, so the participants have no way to determine the reconstruction polynomial $p(x)$ with degree $t - 1$ and they obtain nothing about the shared secrets. For method 2: using NHLR, suppose $m(1 \le m \le t - 1)$ successive participants $\{M_i, M_{i+1}, \ldots, M_{i+m-1}\}$ pool their secret shares, they cannot use m terms $u_{k+i-1}, u_{k+i}, \ldots, u_{k+i+m-2}$ to reveal the constant c, thus they cannot reveal any forward terms u_j for $j < k + i - 1$ or any backward terms u_j for $j > k + i + m - 2$ by using NHLR. Hence, the shared secrets cannot be obtained in such way.

Attack 2: The malicious participant may try to recover the true secrets just by himself.

Discussions: Suppose t arbitrary participants $\{M_i\}_{i\in I}$ pool their shares $\{B_i\}_{i\in I}$. The malicious participant M_j hopes that the honest participants $\{M_i\}_{i\in I\setminus\{j\}}$ will recover some wrong secrets, he provides a wrong share $B_j'(\ne B_j)$. Thus, M_j can get all the true shares $\{B_i\}_{i\in I}$ and recover the true secrets, while the honest participants $\{M_i\}_{i\in\Lambda\setminus\{j\}}$ get shares $\{B_j'\}\cup\{B_i\}_{i\in\Lambda\setminus\{j\}}$ and recover some wrong secrets. However, it is easy to detect whether or not M_j provides a wrong share B_j' because B_j' cannot pass the verification phase.

Attack 3: When one secret sharing process is finished, participant M_j may try to reveal others' secret shadow $s_i(i \neq j)$.

Discussions: $B_i = s_iR_0$. The security of s_i from B_i and R_0 is based on the difficulty of solving the ECDLP. There is no effective method for solving the ECDLP, so this attack does not work.

Attack 4: The dealer D may try to cheat the participants and reveal the secret shadows $s_i(1 \leq i \leq n)$.

Discussions: In our schemes, the secret shadow of each participant is selected by himself, so the dealer is impossible to become a cheater. On the other hand, $B_i = s_iR_0$, D can obtain B_i by computing $B_i = dR_i$ and R_0 by computing $R_0 = dQ$, but it is impossible for D to reveal s_i for the intractability of the ECDLP.

Attack 5: The attacker U may try to reveal d from the public information R_0.

Discussions: $R_0 = dQ$ and the security of d from R_0 and Q is also based on the intractability of the ECDLP, so the attacker U is unable to reveal d.

Attack 6: The attacker U may try to reveal the secret share $B_i(1 \leq i \leq n)$ from the public information R_0.

Discussions: $B_i = s_iR_0$ and U must try to reveal s_i. According to the foregoing discussions, it is impossible for U to reveal s_i, so U is unable to reveal B_i.

Attack 7: The attacker U may try to reveal the coefficients $A_0, A_1, \ldots, A_{t-1}$ from the public information $C_0, C_1, \ldots, C_{t-1}$.

Discussions: $C_i = A_iQ(0 \leq i \leq t-1)$ and the security of A_i from C_i and Q is based on the intractability of the ECDLP, therefore the attacker is unable to reveal the coefficients A_0, $A_1, \ldots, A_{t-1}$.

4.2 Performance Analysis

Verifiability: For some typical schemes, such as [1, 2, 4], there is an assumption that all the participants are honest, but it is so difficult to realize in the real life. For example, a vicious participant may provide a spurious share to other participants in the recovery phase, which will make the vicious participant become the only one who can recover the true secret. It is unfair for other participants who provide correct shares. In order to solve this problem, we design a verification algorithm based on elliptic curve to detect the cheating actions or attacks.

Reuse of the secret shadows: In our schemes, each participant M_i just polls secret share B_i instead of secret shadow s_i to reconstruct the shared secrets and the security of s_i is based on the intractability of the ECDLP. Hence the secret shadow s_i will not be exposed, which makes the reuse of it to be secure.

Public channel: In YCH [4] and SC [5], the dealer randomly selects an integer as secret shadow for each participant and transmits them over secret channels. Instead, the secret shadow of each participant is selected by himself. Hence, secret channels are not necessary and public channels are enough for our schemes.

Reuse of the shared secrets: After an unsuccessful recovery phase in multi-secret sharing schemes [4-7, 10, 11] based on Lagrange interpolation polynomial, the shared secrets must be changed for the security, when $k > t$. Taking ZZZ [6] for example, suppose that the dealer replaces the integer s_0 with s'_0 and keeps the secrets $P_1, P_2, \ldots, P_k$ unchanged, *i.e.*, the reconstruction polynomial $h(x) = P_1 + P_2x + \ldots + P_kx^{k-1} \bmod Q$ keeps unchanged, then each participant M_i has two different pairs (I_i, y_i) and (I'_i, y'_i) of $h(x)$, here $I_i = R_i^{s_0}$, $y_i = h(I_i)$, $I'_i = R_i^{s'_0}$, $y'_i = h(I'_i)$. Thus, any $\lceil t/2 \rceil$ participants can recover the secrets $P_1, P_2, \ldots, P_k$, which goes against the security requirement of a (t, n) threshold secret sharing scheme. In our schemes, it is not necessary to change the shared secrets and the dealer just needs to select another constant c' and integer d'. We can easily see how the terms of NHLR can be changed by choosing new constant and integer. Thus these terms $u_k, \ldots, u_{k+n-1}$ corresponding to participants are changed into $u'_k, \ldots, u'_{k+n-1}$ and the reconstruction polynomial $p(x) = A_0 + A_1x^1 + \ldots + A_{t-1}x^{t-1} \bmod q$ is changed into another polynomial $p'(x)$. So t participants must use their new terms to reconstruct $p'(x)$ or compute c', and the old terms are useless. Therefore, in our schemes the reuse of the shared secrets is secure for the next secret sharing process. We can see the comparison of performance in Table 1.

Table 1. A comparative analysis of the characteristics of our schemes and those in [4-7, 10, 11].

Scheme	YCH[4]	SC[5]	ZZZ[6]	DM[7]	HLC1[10]	LZZ1[11]	our schemes
Verifiability	No	Yes	Yes	Yes	Yes	Yes	Yes
Only public channel is needed	No	No	Yes	Yes	Yes	Yes	Yes
Reuse of the secret shadows is possible	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reuse of the shared secrets is possible	No	No	No	No	No	No	Yes
Time complexity of the recovery phase when $k > t$	$\mathcal{O}(k^2)$	$\mathcal{O}(k^2)$	$\mathcal{O}(k^2)$	$\mathcal{O}(k^2)$	$\mathcal{O}(k^2)$	$\mathcal{O}(k^2)$	$\mathcal{O}(t^2)$

Note: HLC1 [10] denotes the scheme 1 in [10], LZZ1 [11] denotes the scheme 1 in [11], YCH [4], SC [5], ZZZ [6] and DM [7] denote the schemes in [4-7], respectively.

5. CRYPTOGRAPHIC PROPERTIES ANALYSIS

5.1 Public Value

Those schemes [8-13] are based on linear recursion. Here, we compare our schemes with such schemes in terms of public values. As is shown in Table 2, in order to share k multiple secrets with the threshold t, the proposed schemes require $n + k + 2$ public values, whereas DM1 [8] requires $2n + k - t + 1$, both DM2 [8] and HLC2 [10] require $2n + k - t + 2$, LZZ2 [11] requires $2n + k$, DM [12] require $2n + k - t + 4$ and MD [13] require $2n + k - t + 3$. We can easily get the following results, which indicate that our schemes have fewer public values than schemes [8, 10-13].

(1) $n + k + 2 \leq 2n + k - t + 1$ (when $t \neq n$);
(2) $n + k + 2 \leq 2n + k - t + 2$;
(3) $n + k + 2 \leq 2n + k$ (when $n \neq 1$);
(4) $n + k + 2 < 2n + k - t + 4$, $n + k + 2 < 2n + k - t + 3$.

Table 2. Comparison of existing schemes with our schemes.

Scheme	Number of public values	Degree of linear recursion	Degree of reconstruction polynomial	Length of the key in 1024-bit finite fields	Security assumption
DM1 [8]	$2n+k-t+1$	t	$t-1$	1024 bits	DSA
DM2 [8]	$2n+k-t+2$	t	$t-1$	1024 bits	RSA
CYC [9]	$n+k+1$	t	$t-1$	1024 bits	DSA
HLC2 [10]	$2n+k-t+2$	t	$t-1$	340 bits	LFSR
LZZ2 [11]	$2n+k$	t	$t-1$	1024 bits	RSA
DM [12]	$2n+k-t+4$	t	$t+1$	160 bits	ECC
MD [13]	$2n+k-t+3$	$t-1$	$t-1$	340 bits	LFSR
Our schemes	$n+k+2$	$t-1$	$t-1$	160 bits	ECC

Note: DM1 [8] denotes the type1 scheme in [8], DM2 [8] denotes the type2 scheme in [8], HLC2 [10] denotes the scheme 2 in [10], LZZ2 [11] denotes the scheme 2 in [11], CYC [9], DM [12] and MD [13] denote the schemes in [9, 12, 13], respectively.

5.2 Computation Complexity

Here, we compare our schemes with schemes [8-13] in terms of computation complexity.

Construction phase: In the construction phase, the dealer needs to compute linear recursion of degree $t - 1$ in our schemes and MD [13], while the degree of linear recursion is t in schemes [8-12], so the computation complexity of construction phase in our schemes is the same with MD [13], but is lower than schemes [8-12].

Verification phase: Both the verification algorithms in our schemes and DM [12] are based on ECC, while DM1 [8] and CYC [9] are based on DSA, DM2 [8] and LZZ2 [11] are based on RSA, HLC2 [10] and MD [13] are based on LFSR public-key cryptosystem. The computation cost of ECC is much less than that of DSA and RSA. On the other hand, the key length in our schemes is much shorter. Considering a 1024-bit finite field at the same security level, the key length can be represented by 160 bits in our schemes while the key is about six times longer in schemes [8, 9, 11] and two times longer in schemes [10, 13].

Recovery phase: In the recovery phase, the participants should reconstruct a polynomial of degree $t-1$ in our schemes. As is shown in Table 2, the degree of reconstruction polynomial in schemes [8-11, 13] is also $t-1$, whereas it is $t+1$ in DM [12]. On the other hand, in particular case the participants can also use linear recursion of degree $t-1$ to recover the secrets in our schemes and MD [13], whereas it should compute linear recursion of degree t in schemes [8-12]. Therefore, the computation complexity of recovery phase in our schemes is the same with MD [13], but is lower than schemes [8-12].

5.3 Running Time

It is evident that the most time consuming phase in those schemes is the recovery phase. A polynomial of degree n can be constructed in time $\mathcal{O}(n^2)$ by using Lagrange interpolation. In the recovery phase of schemes [4-7, 10, 11], participants need to apply the Lagrange interpolation to construct a polynomial of degree $t-1$ (when $k \leq t$) or $k-1$ (when $k > t$), it can be done in time $\mathcal{O}(t^2)$ (when $k \leq t$) or $\mathcal{O}(k^2)$ (when $k > t$). In our schemes, the secrets can be recovered by the following two methods:

Method 1: Using Lagrange interpolation polynomial.
Method 2: Using NHLR.

Computing NHLR is faster than computing Lagrange interpolation polynomial. For the first method, participants must construct a polynomial of degree $t-1$, it can be done in time $\mathcal{O}(t^2)$. As is shown in Table 1, the recovery phase in our schemes is faster than schemes [4-7, 10, 11] since $\mathcal{O}(t^2) \leq \mathcal{O}(k^2)$, when $k > t$.

6. CONCLUSION

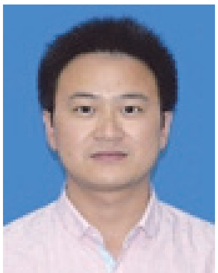
In this paper, we propose two verifiable (t, n) threshold multi-secret sharing schemes based on novel non-homogeneous linear recursions of degree $t-1$. The security and performance analysis of our proposed schemes demonstrate that they allow the reuse of secret shadows and shared secrets, require no secret channels and can withstand seven different attack types. Compared with existing multi-secret sharing schemes [8-13] based on linear recursion, our proposed schemes have fewer public values, lower computation complexity and shorter key length. In addition, the running time of recovery phase in the proposed schemes is less than schemes [4-7, 10, 11].

REFERENCES

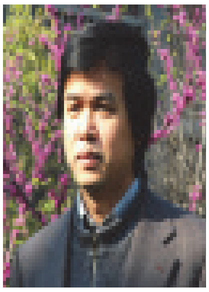
1. G. R. Blakley, "Safeguarding cryptographic key," in *Proceedings of AFIPS National Computer Conference*, 1979, pp. 313-317.
2. A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, 1979, pp. 612-613.
3. H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals*, E83-A(12), 2000, pp. 2762-2765.
4. C. C. Yang, T. Y. Chang, and M. S. Hwang, "A (t, n) multi-secret sharing scheme,"

Applied Mathematics and Computation, Vol. 151, 2004, pp. 483-490.

5. J. Shao and Z. F. Cao, "A new efficient (t, n) verifiable multi-secret sharing (VMSS) based on YCH scheme," *Applied Mathematics and Computation*, Vol. 168, 2005, pp. 135-140.
6. J. J. Zhao, J. Z. Zhang, and R. Zhao, "A practical verifiable multi-secret sharing scheme," *Computer Standards & Interfaces*, Vol. 29, 2007, pp. 138-141.
7. M. H. Dehkordi and S. Mashhadi, "An efficient threshold verifiable multi-secret sharing," *Computer Standards & Interfaces*, Vol. 30, 2008, pp. 187-190.
8. M. H. Dehkordi and S. Mashhadi, "New efficient and practical verifiable multi-secret sharing schemes," *Information Sciences*, Vol. 178, 2008, pp. 2262-2274.
9. Y. K. Chen, J. Yu, X. G. Chen, *et al.*, "Verifiable multi-secret sharing scheme based on homogeneous linear recursion," *Acta Scientiarum Naturalium Universitatis Pekinensis*, Vol. 46, 2010, pp. 709-714.
10. C. Q. Hu, X. F. Liao, and X. Z. Cheng, "Verifiable multi-secret sharing based on LFSR sequences," *Theoretical Computer Science*, Vol. 445, 2012, pp. 52-62.
11. Y. H. Lin, F. T. Zhang, and J. Zhang, "Attacks to some verifiable multi-secret sharing schemes and two improved schemes," *Information Sciences*, Vol. 329, 2016, pp. 524-539.
12. M. H. Dehkordi and S. Mashhadi, "Verifiable secret sharing schemes based on nonhomogeneous linear recursions and elliptic curve," *Computer Communications*, Vol. 31, 2008, pp. 1777-1784.
13. S. Mashhadi and M. H. Dehkordi, "Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem," *Information Sciences*, Vol. 294, 2015, pp. 31-40.
14. M. Tompa, H. Woll, "How to share a secret with cheaters," *Advances in Cryptology-CRYPTO'86*, LNCS 263, 1987, pp. 261-265.
15. Y. L. Tian, J. F. Ma, C. G. Peng, *et al.*, "Fair (t, n) threshold secret sharing scheme," *IET Information Security*, Vol. 7, 2013, pp. 106-112.
16. L. Harn, "Comments on 'Fair (t, n) threshold secret sharing scheme'," *IET Information Security*, Vol. 8, 2014, pp. 303-304.
17. L. Harn, "Secure secret reconstruction and multi-secret sharing schemes with unconditional security," *Security and Communication Networks*, Vol. 7, 2014, pp. 567-573.
18. L. Harn, C. L. Lin, and Y. Li, "Fair secret reconstruction in (t, n) secret sharing," *Journal of Information Security and Applications*, Vol. 23, 2015, pp. 1-7.
19. K. Koyama, U. Maurer, T. Okamoto, *et al.*, "New public-key scheme based on elliptic curves over the ring $\mathbb{Z}_p$," *Advances in Cryptology-Crypto'91*, LNCS 576, 1991, pp. 252-266.
20. N. L. Biggs, *Discrete Mathematics*, Oxford University Press, NY, 1989.



Ben-Hui Zhang (张本慧) received the B.S. degree from Huaibei Normal University, Huaibei, China, in 2008 and the Ph.D. degree from Yangzhou University, Yangzhou, China, in 2013. Now he is a Lecturer in the School of Mathematical Science, Huaibei Normal University, China. His current interests include secret sharing and PEPA model.



Yuan-Sheng Tang (唐元生) received the Ph.D. degree from Nara Institute of Science and Technology, Ikoma, Japan, in 2000. He is a member of IEEE. Now he is a Professor in the School of Mathematical Science, Yangzhou University, China. His research interests include information security and cryptography.