Verifiable secret sharing schemes based on non-homogeneous linear recursions and elliptic curves

Massoud Hadian Dehkordi, Samaneh Mashhadi *

Department of Mathematics, Iran University of Science and Technology, Narmak, Tehran, Iran

Received 31 March 2007; received in revised form 19 November 2007; accepted 25 November 2007

Available online 4 December 2007

Abstract

In this paper, we propose two secure verifiable multi-secret sharing schemes that are based on non-homogeneous linear recursions and elliptic curves over $\mathbb{Z}_N$. Such schemes have simple construction and verification phases. Moreover, these have various techniques for the reconstruction phase. The security of the proposed schemes is based on the security of the ECRSA cryptosystem and the intractability of the ECDLP.

Keywords: Multi-secret sharing; ECRSA cryptosystem; ECDLP; Non-homogeneous linear recursion; Secure channel

1. Introduction

A verifiable secret sharing scheme allows participants to verify the validity of shares of the other participants and her/himself. The verifiable secret sharing scheme plays an important role in design of protocols for secure multi-party computation [3–5,7,9,12,13]. In 1985, the first realization of the verifiable secret sharing scheme was proposed by Chor et al. [3]. Then many literatures [3,11] did several discussions. Thereafter, Harn [6] presented the verifiable multi-secret sharing in 1995. But in his scheme, in order to verify whether the secret shadow is valid, every participant has to check $n!/(n-t)!t!$ equations. In 1997, Chen et al. [2] presented another scheme to improve some drawbacks in [6], but the cost of computing in it is still high. In [9], Shao and Cao (SC) proposed an efficient verifiable multi-secret sharing based on YCH [12] and the intractability of the discrete logarithm problem. It is necessary to mention that the previous schemes need a secure channel between the dealer and participants.

In 2006, Zhao et al. (ZZZ) [13] proposed another practical verifiable multi-secret sharing based on YCH and Hwang–Chang (HC) schemes [12,7]. The RSA cryptosystem [10] is employed in the HC and ZZZ schemes. Hence, a secure channel is unnecessary.

We propose two new efficient and secure verifiable multi-secret sharing schemes based on non-homogeneous linear recursions, the intractability of the ECDLP and the ECRSA cryptosystem, in this paper. We use non-homogeneous linear recursions to give a better performance than the previous schemes. According to the properties of non-homogeneous linear recursions, such schemes have a simple construction and two different methods for the reconstruction phases. Moreover, we employ the ECRSA cryptosystem and a Diffie–Helman key agreement method in these schemes. Therefore, these have a simple verification phase and do not need secure channels. Unlike algorithms that can solve discrete logarithm problem, no efficient method exists for solving the ECDLP, therefore our schemes are more secure than the SC and ZZZ schemes. Analyses show that our schemes are computationally secure and proficient schemes that provide great capabilities for many applications. This paper is organized into seven sections. Section 2 introduces background theories. In Sections 3 and 4, we propose our schemes, whilst Section 5 proposes security analysis. We discuss some important properties of the proposed schemes in Section 6. Finally, we propose conclusions in Section 7.

* Corresponding author. Tel.: +98 2173913400.
E-mail addresses: mhdian@iust.ac.ir (M.H. Dehkordi), smashhadi@mathdep.iust.ac.ir (S. Mashhadi).

0140-3664/S - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.11.014
2. Background theories

2.1. Non-homogeneous linear recursion

In this subsection we introduce the mathematical background of our schemes. A detailed description of non-homogeneous linear recursion can be found in [1].

Definition 2.1. Let \( t \) be a positive integer and \( c_0, c_1, \ldots, c_{t-1}, a_1, a_2, \ldots, a_t \) be real numbers. A non-homogeneous linear recursion of degree \( t \) is defined by the equations

\[ \begin{align*}
\{ u_0 &= c_0, u_1 = c_1, \ldots, u_{t-1} = c_{t-1}, \\
\quad u_{t+j} + a_1 u_{t+j-1} + \cdots + a_t u_j = f(i) \ (i \geq 0),
\end{align*} \]

where \( c_0, c_1, \ldots, c_{t-1} \) and \( a_1, a_2, \ldots, a_t \) are constants.

Theorem 2.2. Let \( F \) be a field, and \( h(x) \) is a polynomial in \( F[x] \). Consider a typical fraction \( h(x)/(1-2x)^m \) where \( x \in F \) and \( \deg h(x) < m \). Then

\[ \frac{h(x)}{(1-2x)^m} = \sum_{i=0}^{\infty} u_i x^i, \]

where \( u_i = p(i)x^i \) and \( p(i) \) is an expression of the form

\[ A_0 + A_1 i + \cdots + A_{m-1} i^{(m-1)}. \]

Proof. [1]. \( \square \)

Lemma 2.3. Suppose the sequence \( (u_i) \) is defined by \([\text{NHLR}]\) as:

\[ \begin{align*}
\{ u_0 &= c_0, u_1 = c_1, \ldots, u_{t-1} = c_{t-1}, \\
\quad \sum_{j=0}^{t} \binom{t}{j} u_{t+j-1} = (-1)^i c \ (i \geq 0),
\end{align*} \]

where \( c_0, c_1, \ldots, c_{t-1} \) and \( c \) are constants. Hence

\( u_i = p(i)(-1)^i \),

where \( p(i) \) is an expression of the form

\[ A_0 + A_1 i + \cdots + A_{t-1} i^{(t-1)}. \]

In other words, \( p(i) \) is a polynomial function of \( i \) with degree at most \( t + 1 \).

Proof. We calculate

\[ \begin{align*}
\sum_{j=0}^{t} \binom{t}{j} x^j \sum_{i=0}^{\infty} u_i x^i &= u_0 + (u_1 + tu_0)x + \cdots \\
+ \left( \sum_{j=0}^{t} \binom{t}{j} u_{t+j-1} \right) x^t + \cdots,
\end{align*} \]

\[ = h_1(x) - cx^{t+1}(1 - 2x + 3x^2 - 4x^3 + \cdots), \]

\[ = h_1(x) - \frac{cx^{t+1}}{1 + x^2}. \]

where \( h_1(x) \) is a polynomial with degree at most \( t - 1 \). So

\[ \sum_{i=0}^{\infty} u_i x^i = \frac{h_1(x)(1 + x^2) - cx^{t+1}}{(1 + x)^2}, \]

and according to Theorem 2.2, we have \( u_i = p(i)(-1)^i \), where \( p(i) \) is a polynomial function of \( i \) with degree at most \( t + 1 \). \( \square \)

Lemma 2.4. Suppose sequence \( (u_i) \) is defined by \([\text{NHLR}]\) as:

\[ \begin{align*}
\{ u_0 &= c_0, u_1 = c_1, \ldots, u_{t-1} = c_{t-1}, \\
\quad \sum_{j=0}^{t} \binom{t}{j} u_{t+j-1} = ci \ (i \geq 0),
\end{align*} \]

where \( c_0, c_1, \ldots, c_{t-1} \) and \( c \) are constants. Then

\( u_i = p(i) \),

where \( p(i) \) is an expression of the form

\[ A_0 + A_1 i + \cdots + A_{t-1} i^{(t-1)}. \]

In other words, \( p(i) \) is a polynomial function of \( i \) with degree at most \( t + 1 \).

Proof. Similarly, we calculate

\[ \begin{align*}
\sum_{j=0}^{t} \binom{t}{j} x^j \sum_{i=0}^{\infty} u_i x^i &= u_0 + (u_1 + tu_0)x + \cdots \\
+ \left( \sum_{j=0}^{t} \binom{t}{j} u_{t+j-1} \right) x^t + \cdots,
\end{align*} \]

\[ = h_2(x) + cx^{t+1}(1 + 2x + 3x^2 + \cdots), \]

\[ = h_2(x) + \frac{cx^{t+1}}{(1 - x)^2}, \]

\[ = \frac{h_2(x)(1 - x)^2 + cx^{t+1}}{(1 - x)^2}, \]

where \( h_2(x) \) is a polynomial with degree at most \( t - 1 \). So

\[ \sum_{i=0}^{\infty} u_i x^i = \frac{h_2(x)(1 - x)^2 + cx^{t+1}}{(1 - x)^{t+2}} \]

and according to Theorem 2.2, we have \( u_i = p(i) \), where \( p(i) \) is a polynomial function of \( i \) with degree at most \( t + 1 \). \( \square \)

2.2. Elliptic curve

In this subsection we review some of the basic definitions and facts about elliptic curves. A detailed description of these can be found in [8].

Definition 2.5. Let \( p \) be a prime greater than 3, and let \( a, b \) be two integers such that \( \gcd(4a^3 + 27b^2, p) = 1 \). An elliptic curve \( E_p(a,b) \) over the prime field \( \mathbb{Z}_p \) is the set of points \( (x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p \) satisfying the Weierstraß equation

\[ y^2 \equiv x^3 + ax + b \pmod{p}, \]

together with the point at infinity \( \mathcal{O}_p \).
The points of the elliptic curve \( E_p(a,b) \) form an abelian group under the tangent-and-chord law defined as follows:

1. \( \mathcal{O}_p \) is the identity element, i.e. \( \forall P \in E_p(a,b), P + \mathcal{O}_p = P \).
2. The inverse of \( P = (x_1,y_1) \) is \( -P = (x_1, -y_1) \).
3. Let \( P = (x_1,y_1) \) and \( Q = (x_2,y_2) \in E_p(a,b) \) with \( P \neq -Q \). Then \( P + Q = (x_3,y_3) \) where
   \[
   x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1,
   \]
   and \( \lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2, \\ \frac{3a^2 + b}{2b} & \text{otherwise.} \end{cases} \]

Lemma 2.6. Let \( \#E_p(a,b) \) denote the order (i.e., the number of points) of the elliptic curve \( E_p(a,b) \) and let \( p \) be an odd prime satisfying \( p \equiv 2 \pmod{3} \). Then, for \( 0 < b < p \), \( E_p(0,b) \) is a cyclic group of order \( \#E_p(0,b) = p + 1 \).

Proof. [8]. \( \square \)

Definition 2.7. Let \( N = p_1p_2 \) with \( p_1 \) and \( p_2 \) two primes greater than 3, and let \( a, b \) be two integers such that \( \gcd(4a^3 + 27b^2, N) = 1 \). An elliptic curve \( E_N(a,b) \) over the ring \( \mathbb{Z}_N \) is the set of points \( (x,y) \in \mathbb{Z}_N \times \mathbb{Z}_N \) satisfying the Weierstrass equation
\[
y^2 \equiv x^3 + ax + b \pmod{N},
\]
together with the point at infinity \( \mathcal{O}_N \).

An addition operation on \( E_N(a,b) \) can be defined in the same way as the addition operation on \( E_p(a,b) \), simply by replacing computations in \( \mathbb{Z}_p \) by computations in \( \mathbb{Z}_N \).

Lemma 2.8. Let \( E_N(a,b) \) be an elliptic curve such that \( \gcd(4a^3 + 27b^2, N) = 1 \) and \( N = p_1p_2 \) (\( p_1, p_2 \): prime). Let \( n_N \) be \( \text{lcm}(\#E_p(1,a), \#E_p(2,a)) \). Then, for any \( P \in E_N(a,b) \) and any integer \( k \),
\[
(kn_N + 1)P = P \pmod{N},
\]
over \( E_N(a,b) \).

Proof. [8]. \( \square \)

Definition 2.9. Let \( P \) and \( Q = tP \) be two points on an elliptic curve. Given \( P \) and \( Q \), one has to recover multiplier \( t \). This problem is defined as the elliptic curve discrete logarithm problem (ECDLP).

Unlike algorithms that can solve discrete logarithm problems, no efficient method exists for solving the ECDLP.

Definition 2.10. An ECRSA cryptosystem contains three phases as follows:

- **Key generation**: User \( U \) chooses large primes \( p_1 \) and \( p_2 \) such that \( p_1 \equiv p_2 \equiv 2 \pmod{3} \). \( U \) computes the product \( N = p_1p_2 \) and \( n_N = \text{lcm}(p_1 + 1, p_2 + 1) \). \( U \) chooses an integer \( e \) which is coprime to \( n_N \), and computes an integer \( d \) such that \( ed \equiv de \equiv 1 \pmod{n_N} \). Summarizing, \( U \)’s secret key is \( d \) and \( U \)’s public key is \( N, e \).
- **Encryption procedure**: A plaintext \( M = (m_s, m_r) \) is an integer pair, where \( m_s, m_r \in \mathbb{Z}_N \). Let \( M = (m_s, m_r) \) be a point on the elliptic curve \( E_N(0,b) \), where \( b \) is determined by \( m_s \) and \( m_r \). Sender \( A \) encrypts the point \( M \) by encryption function \( E(.) \) with the receiver’s public key \( e \) and \( N \) as
\[
C = E(M) = eM \pmod{N},
\]
and sends a ciphertext pair \( C = (c_s, c_r) \) to a receiver \( B \).
- **Decryption procedure**: Receiver \( B \) decrypts a point \( C \) by the decryption function \( D(.) \) with his/her secret key \( d \) and public key \( N \) as
\[
M = D(C) = dC \pmod{N}. 
\]

Theorem 2.11 proves that how \( B \) decrypts the ciphertext \( C \).

Theorem 2.11. Let \( N = p_1p_2 \), \( n_N = \text{lcm}(p_1 + 1, p_2 + 1) \) and \( de \equiv 1 \pmod{n_N} \). If \( C = eM \pmod{N} \), then \( dC = M \pmod{N} \).

Proof. Let \( dC = d(eM) = dM \pmod{N} \) and \( N = p_1p_2 \).

3. Type 1 scheme

In this section we propose a new scheme that is based on the non-homogeneous linear recursions.

3.1. Initialization phase

\( P_1, P_2, \ldots, P_k \) denote \( k \) secrets to be shared among \( n \) participants. The notations \( N, p_1, p_2, n_N \) in this scheme are the same as those in the ECRSA cryptosystem. We consider the [NHLP] which is defined by Eqs. (**) in Lemma 2.4.

Let \( q \) be a prime number, such that \( q > c, \quad \binom{t}{i} \) for \( i = 1, 2, \ldots, t \). The dealer \( D \) first selects two large prime numbers, \( p_1 \) and \( p_2 \), and computes \( N = p_1p_2 \). \( D \) also considers \( Q \in E_N(0,b) \) such that the discrete logarithm problem is infeasible in cyclic subgroup \( \langle Q \rangle \). \( D \) publishes \( \{N, Q\} \).

Each participant \( M_i \), randomly selects an integer \( s_i \), as her/his own secret shadow and computes \( R_i = s_iQ \). Then \( M_i \) transmits \( (R_i, i) \) to the dealer \( D \). \( D \) has to ensure that \( R_i \neq R_j \) for all \( i \neq j \) to keep different participants from using the same secret shadow. Once \( R_i = R_j \), \( D \) should demand these participants to select different secret shadows until \( R_i \)’s are different for \( i = 1, 2, \ldots, n \). The dealer publishes \( (R_1, R_2, \ldots, R_n) \).
3.2. Construction phase

The dealer D performs the following steps:

(1) Randomly select an integer \( e \) such that \( \gcd(e, n_N) = 1 \) and compute \( d \) such that \( ed \equiv de \equiv 1 \pmod{n_N} \). \(^1\)
(2) Compute \( R_0 := dQ \) and \( B_i := dR_i \) over \( E_N(0, b) \) for \( i = 1, 2, \ldots, n \).
(3) Compute \( I_i = x_{B_i} + y_{B_i} \) for \( i = 1, 2, \ldots, n \), where \( x_{B_i} \) is the \( x \)-coordinate of the point \( B_i \) and \( y_{B_i} \) is the \( y \)-coordinate of the point \( B_i \).
(4) Consider [NHLR] which is defined by the equations

\[
\begin{align*}
    u_0 &= I_1, \quad u_1 = I_2, \ldots, u_{n-1} = I_i, \\
    \sum_{j=0}^{t} (-1)^i u_{i+j} &= c \mod q \quad (i \geq 0), \quad c \in \mathbb{Z}.
\end{align*}
\]

(5) Compute \( u_t \), \( t \leq i \leq n + k + 3 \).
(6) Compute \( y_i = I_i - u_{i-1} \), for \( t < i \leq n \) and \( r_j = P_i - u_{i+j} \), for \( 1 \leq i \leq k \).
(7) Publish 

\[
(R_0, e, r_1, r_2, \ldots, r_k, y_{i+1}, y_{i+2}, \ldots, y_n, u_{n+k+2}, u_{n+k+3}).
\]

In the initialization protocol, each participant chooses her/his shadow by her/himself, so that it is absolutely impossible for the dealer to become a cheater.

3.3. Verification phase

At first we show that how each participant \( M_i \) can compute her/his secret share \( B_i \).

**Theorem 3.1.** Each participant \( M_i \) can compute \( s_i R_0 \) to gain her/his secret share \( B_i \).

**Proof.**

\( s_i R_0 = s_i dQ = ds_i Q = dR_i = B_i. \)

Without loss of generality, suppose at least \( t \) participants \( \{M_j\}_{j=1}^t \) pool their shares \( \{B_j\}_{j=1}^t \) to recover the secrets \( P_1, P_2, \ldots, P_k \). Participant \( M_i \) can check whether others’ secret shadows are valid by the following equations:

\[ eB_j = R_j \quad \text{over } E_N(0, b) \quad j = 1, 2, \ldots, t, \quad j \neq i. \]

**Theorem 3.2.** Let participant \( M_j \) provide \( B_j \). If \( eB_j = R_j \) over \( E_N(0, b) \), then \( B_j \) is true, otherwise \( B_j \) is false.

**Proof.**

\[ eB_j = edR_j, \]
\[
= (kn_N + 1)R_j \quad ed \equiv 1 \pmod{n_N}, \]
\[
= R_j \quad \text{Lemma 2.8.} \]

3.4. Recovery phase

Here we propose two different techniques for the reconstruction phase.

3.4.1 Suppose \( t \) arbitrary participants \( \{M_i\}_{i \in I} \) pool their secret shares \( \{B_i\}_{i \in I} \). They perform the following steps:

(1) Compute \( I_i = x_{B_i} + y_{B_i} \) for \( i \in I \).
(2) Compute \( t \) terms \( u_{i-1} \)’s of [NHLR] for \( i \in I \) as:

\[
u_{i-1} = \begin{cases} I_i & \text{if } 1 \leq i \leq t, \\
I_i - y_i & \text{if } t < i \leq n. \end{cases}
\]

(3) Use \( t + 2 \) pairs \( (i - 1, u_{i-1}) \)’s where \( i \in I \), \( (n + k + 2, u_{n+k+2}) \), and \( (n + k + 3, u_{n+k+3}) \) to construct the \( (t + 1) \)th degree polynomial \( p(x) \mod q \):

\[
p(x) = \sum_{i \in I} Y_i \prod_{j \neq i} \frac{x - X_j}{X_i - X_j} \mod q,
\]

\[
= A_0 + A_1 x^1 + \cdots + A_t x^t \mod q.
\]

(4) Compute \( u_j = p(j) \) for \( j = n + 1, n + 2, \ldots, n + k \).
(5) Recover \( P_j = u_{j+n} + r_j \) for \( j = 1, 2, \ldots, k \).

3.4.2 Suppose \( t \) participants \( \{M_i, M_{i+1}, \ldots, M_{i+t-1}\}, \ (1 \leq i \leq n - i + 1) \) pool their secret shares \( \{B_j\}_{j=i}^{i+t-1} \). In other words, suppose the \( t \) participants indexes are successive. Besides previous methods proposed in 3.4.1, they can perform the following steps:

(1) Compute \( I_j = x_{B_j} + y_{B_j} \) for \( j = i, i+1, \ldots, i+t-1 \).
(2) Compute \( t \) terms \( u_{j-1} \)’s of [NHLR] for \( j = i, i+1, \ldots, i+t-1 \) as:

\[
u_{j-1} = \begin{cases} I_j & \text{if } 1 \leq j \leq t, \\
I_j - y_j & \text{if } t < j \leq n. \end{cases}
\]

(3) Compute terms \( u_j \)’s of [NHLR] for \( j = i+t, i+t, \ldots, n+k \) as:

\[
u_{m+t} = cm + \sum_{j=1}^{t} (-1)^{j+1} u_{m+j-1} \mod q,
\]

for \( m = i - 1, i, \ldots, k + n - t \).

(4) Recover the shared secrets \( P_j = u_{j+n} + r_j \) for \( j = 1, 2, \ldots, k \).

4. Type 2 scheme

In this section we propose another scheme that is based on the non-homogeneous linear recursion. Other schemes can be defined similarly.

\[1^1 \] We use \( (X_i, Y_i) \) for \( i \in I' \) where \( I' = I \cup \{n + k + 2, n + k + 3\} \) to denote these \( t + 2 \) pairs, respectively.
4.1. Initialization phase

$P_1, P_2, \ldots, P_k$ denote $k$ secrets to be shared among $n$ participants. The notations $N, p_1, p_2, q_1$ in this scheme are the same as those in the ECRSA cryptosystem. We consider the [NHLR] which is defined by Eqs. [9] in Lemma 2.3. Let $q$ be a prime number, such that $q > c$, $\left(\frac{t}{i}\right)$ for $i = 1, 2, \ldots, t$. The dealer D first selects two large prime numbers, $p_1$ and $p_2$, and computes $N = p_1p_2$. D also considers $Q \in E_N(0, b)$ such that the discrete logarithm problem is infeasible in cyclic subgroup $\langle Q \rangle$. D publishes $\{N, Q\}$.

Each participant $M_i$ randomly selects an integer $s_i$ as her/his own secret shadow and computes $R_i = s_iQ$. Then $M_i$ transmits $(R_i, i)$ to the dealer D. D has to ensure that $R_i \neq R_j$ for all $i \neq j$ to keep different participants from using the same secret shadow. Once $R_i = R_j$, D should demand these participants to select different secret shadows until $R_i$'s are different for $i = 1, 2, \ldots, n$. The dealer publishes $(R_1, R_2, \ldots, R_n)$.

4.2. Construction phase

The dealer D performs the following steps:

1. Randomly select an integer $e$ such that $\gcd(e, n_N) = 1$ and compute $d$ such that $ed \equiv 1 \pmod{n_N}$.
2. Compute $R_0 := dQ$ and secret shares $B_i := dR_i$ over $E_N(0, b)$ for $i = 1, 2, \ldots, n$.
3. Compute $I_i = x_{B_i} + y_{B_i}$ for $i = 1, 2, \ldots, n$, where $x_{B_i}$ is the $x$-coordinate of the point $B_i$ and $y_{B_i}$ is the $y$-coordinate of the point $B_i$.
4. Consider [NHLR] which is defined by the equations
   \[
   \begin{align*}
   u_0 &= I_1, u_1 = I_2, \ldots, u_{i-1} = I_i, \\
   \sum_{j=0}^{t} \left(\frac{t}{i} \right) u_{i-j} &= (-1)^{ci} \bmod q \quad (i \geq 0), \quad c \in \mathbb{Z}.
   \end{align*}
   \]
5. Compute $u_i$, $t \leq i \leq n + k + 3$.
6. Compute $y_i = I_j - u_{i-1}$, for $t < i \leq n$ and $r_i = P_j - u_{i+n}$, for $1 \leq i \leq k$.
7. Publish $(R_0, e, r_1, r_2, \ldots, r_k, y_{i+1}, y_{i+2}, \ldots, y_{n}, u_{n+k+2}, u_{n+k+3})$.

In the initialization protocol, each participant chooses her/his shadow by her/himself, so that it is absolutely impossible for the dealer to become a cheater.

4.3. Verification phase

Each participant $M_i$ can compute $s_iR_0$ to gain her/his secret share $B_i$. Without loss of generality, suppose at least $t$ participants $\{M_i\}_{i=1}^t$ pool their secrets $\{B_i\}_{i=1}^t$ to recover the secrets $P_1, P_2, \ldots, P_k$. Participant $M_t$ can check whether others’ secret shadows are valid by the following equations:

\[
eB_j = R_j \quad \text{over } E_N(0, b) \quad j = 1, 2, \ldots, t, \quad j \neq t.
\]

4.4. Recovery phase

Now we propose two different ways for the recovery phase.

4.4.1

Suppose $t$ arbitrary participants $\{M_i\}_{i \in \mathbb{I}}$ pool their secret shares $\{B_i\}_{i \in \mathbb{I}}$. They perform the following steps:

1. Compute $I_i = x_{B_i} + y_{B_i}$ for $i \in \mathbb{I}$.
2. Compute $t$ terms $u_{i-1}$'s of [NHLR] for $i \in \mathbb{I}$ as:
   \[
   u_{i-1} = \begin{cases} 
   I_i & \text{if } 1 \leq i \leq t, \\
   I_i - y_i & \text{if } t < i \leq n.
   \end{cases}
   \]
3. Use $t + 2$ pairs $i, (-1)^{i-1}u_{i-1}$'s where $i \in \mathbb{I}, (n + k + 2, (-1)^{i+k+2}u_{n+k+2})$ and $(n + k + 3, (-1)^{i+k+3}u_{n+k+3})$ to construct the $(t + 1)$th degree polynomial $p(x) \bmod q$:
   \[
   p(x) = \sum_{i \not\in \mathbb{I}} y_i \prod_{j \not\in \mathbb{I}, j \neq i} \frac{x - X_j}{X_i - X_j} \bmod q,
   \]
   \[
   = A_0 + A_1x^1 + \cdots + A_{t+1}x^{t+1} \bmod q.
   \]
4. Compute $u_i = (-1)^{t}p(j) \bmod q$ for $j = n + 1, n + 2, \ldots, n + k$.
5. Recover $P_j = u_{j+n} + r_j$ for $j = 1, 2, \ldots, k$.

4.4.2

Suppose $t$ participants $\{M_i, M_{i+1}, \ldots, M_{i+t-1}\}$, $(1 \leq i \leq n - t + 1)$ pool their secret shares $\{B_j\}_{j \in \mathbb{I}}$. Besides previous methods proposed in 4.4.1, they can perform the following steps:

1. Compute $I_j = x_{B_j} + y_{B_j}$ for $j = i, i+1, \ldots, i+t-1$.
2. Compute $t$ terms $u_{i-1}$'s of [NHLR] for $j = i, i+1, \ldots, i+t-1$, as:
   \[
   u_{j+1} = \begin{cases} 
   I_j & \text{if } 1 \leq j \leq t, \\
   I_j - y_j & \text{if } t < j \leq n.
   \end{cases}
   \]
3. Compute terms $u_j$'s of [NHLR] for $j = i + t - 1, i + t, \ldots, n + k$ as:
   \[
   u_{n+t} = (-1)^{m}c_{m} - \sum_{j=1}^{t} \left(\frac{t}{i} \right) u_{i+t-j} \bmod q
   \]
   \[
   \text{for } m = i - 1, i, \ldots, k + n - t.
   \]
4. Recover $P_j = u_{j+n} + r_j$ for $j = 1, 2, \ldots, k$.

We use $(X_i, Y_i)$ for $i \in \mathbb{I}'$, where $\mathbb{I}' = \mathbb{I} \cup \{n + k + 2, n + k + 3\}$ to denote these $t + 2$ pairs, respectively.
5. Security analysis

In this section, we prove that our schemes are multi-use schemes and then analyze the security of them.

5.1. Multi-use scheme

In our schemes each participant $M_i$ just polls his/her secret share $B_i$ in the verification and recovery phases. The secret share $B_i$ is computed by the formula $B_i = s_i R_0$. The security of $s_i$ from $R_0$ and $B_i$ is based on the difficulty of the ECDLP. Therefore, the secret shadow $s_i$ will not be disclosed and the reuse of it is secure.

5.2. Security analyses

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

- Analysis: In these schemes the shared secrets are reconstructed by using one of the following methods:
  1. Using the Lagrange interpolation polynomial (3.4.1 and 4.4.1).
  2. Using the non-homogeneous linear recursion of degree $t$ (3.4.2 and 4.4.2).

In each of these methods, participants must compute $t$ terms of the $j$th degree polynomial $p(x)$ to compute these terms. Suppose $m$ arbitrary participants $\{ M_j \}_{j \in I}$ where $1 \leq m < t$ pool their secret shares. Besides the two public pairs, they can obtain $m$ pairs $(j - 1, u_{j-1})$'s in the Type 1 scheme (3.4.1). Similarly they can obtain $m$ pairs $(j - 1, (-1)^{j-1} u_{j-1})$'s in the Type 2 scheme (4.4.1). Hence the number of obtained pairs is less than $t + 1$ in each of these schemes. Therefore, these participants can have no way of determining the $(t + 1)$th degree polynomial $p(x)$, nor can they derive anything about the terms $u_j$'s for $1 \leq j \leq n + k$ or shared secrets in each of these schemes. Each term $u_i$ of [NHLR] depends on the previous $t$ terms. Hence, in the second method participants must compute $t$ terms $u_j$'s for $n - t < j \leq n$ to gain forward terms $u_j$'s for $n + 1 \leq j \leq n + k$. Suppose $m$ participants $\{ M_i, M_{i+1}, \ldots, M_{i+m-1} \}$ where $(1 \leq i \leq n + 1 - m, 1 \leq m < t)$ pool their secret shares they can gain $m$ terms $u_j$, $(1 \leq m \leq t, i - 1 < j \leq i + m - 2)$ in (3.4.2 or 4.4.2). Hence they are unable to reveal any previous terms $u_j$'s for $j < i - 1$ or any forward terms $u_j$’s for $j > i + m - 2$ by using [NHLR] in these schemes. Therefore, the shared secrets cannot be obtained by this method.

5.2.2 Attack. A plotter $E$ may try to reveal a secret share $s_i$ of participant $M_i$ from public information $R_i$.

- Analysis: If $R_i$ is computed by the formula $R_i = s_i Q$. Hence the security of $s_i$ from $R_i$ and $Q$ is based on the difficulty of solving the ECDLP. No efficient method exists for solving the ECDLP, hence $s_i$ is secure.

5.2.3 Attack. After the recovery phase, participant $M_i$ may try to reveal another’s secret shadow $s_j$ where $1 \leq j \leq n$.

- Analysis: $B_i = s_i R_0$ and the security of $s_j$ from $B_i$ and $R_0$ is based on the intractability of the ECDLP. Hence this attack does not work against our schemes.

5.2.4 Attack. The dealer may try to reveal the secret shadows $s_i$'s for $1 \leq i \leq n$.

- Analysis: The dealer can compute $B_i = dR_i$. But $B_i = s_i R_0$, hence the security of each $s_i$ from $B_i$ and $R_0$ is based on the intractability of the ECDLP. Therefore $s_i$’s are secure.

5.2.5 Attack. The plotter $E$ may try to reveal $d$ from the $R_0$ or $e$.

- Analysis: $R_0 = dQ$ and $ed = 1 \pmod{n_0}$. Therefore, the security of $d$ from the $R_0$ and $Q$ is based on the difficulty of solving the ECDLP. Moreover the security of $d$ from $e$ is based on the security of the ECRSA cryptosystem. Therefore the plotter is unable to reveal $d$.

5.2.6 Attack. The plotter $E$ may try to reveal a secret share $B_i$ from the public information $R_i$ and $R_0$.

- Analysis: $B_i = dR_i = s_i R_0$. Therefore, $E$ must try to reveal $s_i$ or $d$. According to the previous analyses $E$ is unable to reveal them.

6. Discussions

In this section, we compare our schemes with the SC and ZZZ schemes in terms of: public values, computational complexity, running time and public channel.

6.1. Public values

In order to share $k$ secrets, we have $2(n + 2) + k - t$ public values in the proposed schemes, whereas the ZZZ
scheme requires $2(n + 1)$ (when $k \leq t$) or $2(n + 1) + k - t$ (when $k > t$). Also, the SC scheme has $(n + t + 1)$ (when $k \leq t$) or $2k + n - t + 1$ (when $k > t$) public values. Thus, our schemes have the same number of public values as the ZZZ secret sharing when $k \geq t$. Furthermore, it is clear that the number of public values in these schemes is fewer than that in the ZZZ scheme when $2 + k \leq t$. As compared with the SC scheme, it is easy to see that $2(n + 2) + k - t \leq n + t + 1$ where $n - t \leq t - (k + 3)$, and $2(n + 2) + k - t \leq 2k + n - t + 1$, where $n + 3 \leq k$. Therefore, the proposed schemes become more attractive and practical, especially when we have one of the following cases:

- $n - t \leq t - (k + 3)$.
- The number of secrets $k$ is more than the number of participants $n$.

In these cases, as compared with the SC and ZZZ schemes, these schemes do not require a secure channel, neither do they need more public values.

6.2. Computational complexity

- **Initialization phase:** In the SC scheme, the dealer chooses and distributes secret shadows to each participant by a secure channel, whilst the ZZZ and our schemes do not need a secure channel at all, but instead use some computations. In the ZZZ scheme, each participant must compute one exponentiation in the module $N$, whilst in our schemes, the verifier must compute a multiple $sB$ of a point $B \in E_n(0,b)$. Therefore, the initialization phase in our schemes is easier than that in the ZZZ scheme because multiplication of a number and an elliptic curve point is easier than a modular exponentiation. Moreover, these schemes are multi-use schemes, hence participants do not need to repeat these computations in the next secret sharing schemes.

- **Construction phase:** The construction phase of the ZZZ and SC schemes is the same as that of YCH secret sharing. In other words, these schemes employ the polynomials of degree $(t - 1)$ or $(k - 1)$ to distribute secrets. But in our schemes, the secrets are distributed only by using NHLR. Furthermore, in the ZZZ scheme the dealer $D$ must compute modular exponentiations whilst $D$ must compute multiples of an elliptic curve point in our schemes. Therefore, our secret constructions are easier and faster than previous schemes because, using the linear recurrences and computing multiples of an elliptic curve point are easier than using the polynomials and computing modular exponentiations, respectively. Moreover, the construction phase in ZZZ and SC schemes has two different implementation in two different cases, which makes these secret sharing more complex than ours.

- **Recovery phase:** In the ZZZ and SC schemes, the secrets are reconstructed only by using the Lagrange interpolation polynomial, whereas our schemes have various methods for the secret reconstruction. Furthermore, in the ZZZ and SC schemes, the secret reconstruction has two different implementations in two separate cases, which makes these schemes more complex than ours. In other words, participants must reconstruct a $(t - 1)$th degree polynomial when $k \leq t$ and a $(k - 1)$th degree polynomial when $k > t$. In our schemes, the secrets are recovered only by reconstructing a $(t + 1)$ degree polynomial. Also, in particular cases, we can use NHLR to reconstruct secrets, making it very easy and fast.

- **Verification phase:** In the ZZZ scheme in order to verify a secret share $I_i$ provided by $M_i$, the verifier must compute one exponentiation in the module $N$, whilst in our schemes, the verifier must compute a multiple $eB$ of a point $B \in E_n(0,b)$. Therefore, the verification phase in our schemes is easier than that in the ZZZ scheme. In the ZZZ and our schemes, each participant chooses her/his shadow. Hence, it is impossible for the dealer to cheat. In the SC scheme, the verifier must compute some power products, thus compared with the ZZZ and our schemes the verification in the SC secret sharing is more complex [4]. Moreover, in the SC scheme each participant must check whether her/his shadow is valid.

6.3. Running time

It is evident that the most time consuming phase in these schemes is the recovery phase. In the SC and ZZZ schemes, participants use the Lagrange interpolation polynomial to redistribute secrets. A nth degree polynomial can be constructed in time $O(n^2)$ by using the Lagrange interpolation, so the recovery phase can be done in time $O(t^2)$ (when $k \leq t$) or $O(k^2)$ (when $k > t$) in the SC and ZZZ schemes. The recovery phase of our schemes is based on one of the following methods:

1. Using the Lagrange interpolation polynomial.
2. Using the [NHLR] of degree $t$.

Computing the linear recurrences is faster than the first method. In the first way, we must constructing a $(t + 1)$th degree polynomial. According to the previous discussion, it can be done in time $O((t + 1)^2) = O(t^2)$. Therefore the recovery phase in our schemes can be done in time $O(t^2)$.

6.4. Public channel

In the SC scheme, the dealer distributes each secret shadow $s_i$ to each participant $M_i$ over a secure channel. Instead, the ZZZ and our schemes employ some public key cryptographic process, such as the Diffie–Hellman pro-
cess and ECRSA cryptosystem. Therefore, a secure channel is not necessary at all in these schemes.

7. Conclusions

In this paper, we have proposed two new verifiable multi-secret sharing schemes based on the non-homogenous linear recursions. These schemes have simple construction and verification phases. Moreover, each of such schemes has two different methods for the recovery phase. As compared with the SC and ZZZ schemes, ours are more efficient and secure. Analyses show that these are computationally secure and efficient schemes and provide many functions for practical applications. In addition, a secure channel is not necessary at all. These schemes have the following properties:

1. The secret shadow of each participant can be reused after the secrets are recovered.
2. Each participant holds only one shadow while sharing many secrets with other participants.
3. The size of each shadow is as short as that of each shared secret.
4. Each participant selects her/his secret shadow by her/himself and the dealer does not know the shadow of any participant.
5. Participants do not need to follow a specific order to recover the secrets.
6. Each participant is able to check whether another participant provides the true information or not.
7. It is absolutely impossible for the dealer to become a cheater.
8. The secrets are recovered parallelly.
9. No secret communication exists between dealer and participants and a secure channel is unnecessary.
10. These scheme use of elliptic curve so are more secure and efficient than the previous schemes.
11. The security of them is based on the security of the ECRSA cryptosystem and the intractability of the ECDLP.
12. They have simple construction and verification phases.
13. Each of these schemes has two different methods for the recovery phase.

References