



# Verifiable multi-secret sharing based on LFSR sequences

Chunqiang Hu<sup>a,b,\*</sup>, Xiaofeng Liao<sup>a</sup>, Xiuzhen Cheng<sup>b</sup>

<sup>a</sup> State Key Lab. of Power Transmission Equipment & System Security and New Technology, College of Computer Science, Chongqing University, Chongqing 400030, China

<sup>b</sup> Department of Computer Science, The George Washington University, Washington DC 20052, USA

## ARTICLE INFO

### Article history:

Received 1 August 2011  
Received in revised form 29 November 2011  
Accepted 3 May 2012  
Communicated by D.-Z. Du

### Keywords:

Verifiable multi-secret sharing  
LFSR-based public key cryptosystem  
Threshold scheme  
Cryptography

## ABSTRACT

In verifiable multi-secret sharing schemes (VMSSs), many secrets can be shared but only one share is kept by each user and this share is verifiable by others. In this paper, we propose two secure, efficient, and verifiable  $(t, n)$  multi-secret sharing schemes, namely Scheme-I and Scheme-II. Scheme-I is based on the Lagrange interpolating polynomial and the LFSR-based public key cryptosystem. The Lagrange interpolating polynomial is used to split and reconstruct the secrets and the LFSR-based public key cryptosystem is employed to verify the validity of the data. Scheme-II is designed according to the LFSR sequence and the LFSR-based public key cryptosystem. We compare our schemes with the state-of-the-art in terms of attack resistance, computation complexity, and so on, and conclude that our schemes have better performance and incur less computation overhead. Our schemes can effectively detect a variety of forgery or cheating actions to ensure that the recovery of the secrets is secure and credible, and the length of the private key is only one third of that of others for the same security level.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

Secret sharing has become one of the most important research areas in modern cryptography since it was proposed in 1979 [1,2]. Nowadays it has been widely adopted by many emerging applications, including opening a bank vault, launching a nuclear attack, transferring electronic funds, to name a few. Secret sharing plays a significant role in protecting secret information from becoming lost, being destroyed/alterred, or falling into the wrong hands [3,4].

Several drawbacks of the original  $(t, n)$ -threshold secret sharing schemes [1,2] have been identified [5], which are listed as follows:

1. Each secret sharing process involves only one to-be-shared secret.
2. The secret shares can be used only once. After a secret is recovered, the dealer must redistribute a fresh share, also known as a *shadow*, over a secure channel to every participant.
3. These schemes assume that the dealer and the participants are honest. Nevertheless, a dishonest dealer may distribute a fake shadow to a certain participant, and a malicious participant may provide a fake share to other participants. Such behaviors can significantly affect the effectiveness of secret sharing schemes.

To overcome the first two drawbacks, multi-secret sharing (MSS) was proposed [6–9]. Such a scheme requires that multiple secrets are shared and each participant holds one share of each secret. For example, the multistage secret sharing scheme proposed in [6] employs a one-way function to overcome the second drawback, while the YCH scheme [9] utilizes a two-variable one-way function based Shamir's secret sharing [1] to tackle both drawbacks. Although MSS schemes have

\* Corresponding author at: Department of Computer Science, The George Washington University, Washington DC 20052, USA. Tel.: +1 202 294 7334.  
E-mail addresses: [chu@gwmail.gwu.edu](mailto:chu@gwmail.gwu.edu), [chu@gwu.edu](mailto:chu@gwu.edu), [hcq0394@163.com](mailto:hcq0394@163.com) (C. Hu).

many useful applications [10–13], they cannot verify the honesty of either the dealer or the participants, which means that they cannot handle the scenarios where dishonest dealers and/or participants exist.

The third drawback is tackled by adding the concept of verifiability. In verifiable multi-secret sharing (VMSS), the validity of the shares can be verified; hence dishonesty of the participants and/or the dealer can be identified. The realization of VMSS was first presented by Chor et al. [5] in 1985, and then further investigated by many other researchers [14–17]. In 2005, Shao and Cao [18] introduced the discrete logarithm (DL) into the YCH scheme [9], which is believed to be relatively efficient but does not have the property of verification, to yield an efficient verifiable multi-secret sharing. This new scheme requires the dealer to distribute one secret shadow to each participant over a secure channel. Later, in 2007, Zhao et al. [19] proposed a practical VMSS termed ZZZ, which combines the YCH scheme with the Hwang–Chang (HC) scheme [20] to employ the RSA cryptosystem and the Diffie-Helman key agreement mechanism. The ZZZ scheme does not need a secure channel, but this benefit comes with a high computation overhead. In 2008, Dehkordi and Mashhadi [13,21] presented two efficient VMSS schemes, which employ the intractability of the discrete logarithm and the RSA cryptosystem to improve the YCH scheme. But these two schemes are too resource-intensive, resulting in lower speeds. In 2010, a new VMSS scheme based on the cellular automata theory [22] was proposed. This scheme has a linear computational complexity, but it fails to resist certain attacks such as the conspiracy attack, as analyzed in this paper.

We present two novel efficient VMSS schemes in this paper, denoted by Scheme-I and Scheme-II, to overcome the three drawbacks mentioned above. To split and reconstruct the secrets, Scheme-I adopts the Lagrange interpolating polynomial while Scheme-II utilizes the homogeneous LFSR sequence. The verification phases of both schemes exploit the LFSR public key cryptosystem and the LFSR-based public key distribution [23,24]. Our analysis indicates that these two schemes are computationally secure and efficient. For the same strength of security, the lengths of the private keys obtained from our schemes are only one-third of those computed by others. Note that the validation of Scheme-I and Scheme-II are established on the different theories in comparison with the previously proposed schemes.

The structure of the paper is organized as follows. In the next section, the homogeneous LFSR sequence and the third-order LFSR are briefly introduced. A review of the YCH scheme is also given in this section. We propose our schemes and analyze their feasibility in Section 3. Sections 4 and 5 detail our security analysis and performance analysis, respectively. We conclude this paper in Section 6.

## 2. Preliminaries

In this section, we briefly introduce the homogeneous LFSR sequence, the third-order LFSR, the LFSR public-key cryptosystem, and the YCH scheme.

### 2.1. Homogeneous LFSR Sequence

Let  $F = GF(p)$ , where  $p$  is a prime. Denote by  $f(x) = x^t - c_1x^{t-1} - \dots - c_{t-1}x - c_t$  a polynomial over  $F$ , with  $c_1, c_2, \dots, c_t$  being constants in  $F$ . We say that  $\mathbf{s} = \{s_k \mid k = 0, 1, 2, \dots\}$  is a homogeneous LFSR sequence of order  $t$  (generated by  $f(x)$ ) if  $\mathbf{s}$  satisfies the following linear recursive relation:

$$s_{j+t} = \sum_{i=1}^t c_i s_{j+t-i}, \quad j = 0, 1, \dots \tag{1}$$

The vector  $(s_k, s_{k+1}, \dots, s_{k+t-1})$ , which contains  $t$  consecutive terms of  $\mathbf{s}$ , is called the  $k$ -th state of  $\mathbf{s}$ , denoted by  $\mathbf{s}_k$ . The initial state of  $\mathbf{s}$  is  $(s_0, s_1, s_2, \dots, s_{t-1})$ .

From basic field theory,

$$x^t = c_1x^{t-1} + c_2x^{t-2} + \dots + c_{t-1}x + c_t \tag{2}$$

is called the characteristic equation of Eq. (1). The roots of Eq. (2) are called the characteristic roots of Eq. (1). We assume that Eq. (2) has  $t$  characteristic roots, and these roots do not have to be distinct. Let  $x_1, x_2, \dots, x_t$  be the distinct characteristic roots, with multiplicities  $m_1, m_2, \dots, m_t$ , respectively, such that  $\sum_{i=1}^t m_i = t$ . Then Eq. (2) can be presented by

$$(x - x_1)^{m_1} (x - x_2)^{m_2} \dots (x - x_t)^{m_t} = 0. \tag{3}$$

We are particularly interested in the sequence  $u_{j+t} = c_1u_{j+t-1} + c_2u_{j+t-2} + \dots + c_tu_j, j = 0, 1, \dots$ . Some properties of its characteristic equation are summarized by the following two theorems.

**Theorem 1.** *If the characteristic equation of the sequence  $u_i$  has distinct characteristic roots  $x_1, x_2, \dots, x_t$ , we have*

$$u_i = A_1x_1^i + A_2x_2^i + \dots + A_t x_t^i, \tag{4}$$

where  $A_1, A_2, \dots, A_t$  are constants that can be computed from  $c_1, c_2, \dots, c_t$ .

**Proof.** Let  $f(x) = u_0 + u_1x + \dots + u_{t-1}x^{t-1} + u_t x^t + \dots$ . Then

$$\begin{aligned} f(x) &= u_0 + u_1x + \dots + u_{t-1}x^{t-1} + (c_1u_{t-1} + c_2u_{t-2} + \dots + c_tu_0)x^t + (c_1u_t \\ &\quad + c_2u_{t-1} + \dots + c_tu_1)x^{t+1} + (c_1u_{t+1} + c_2u_{t+2} + \dots + c_tu_2)x^{t+2} + \dots \\ &= c_t x^t (u_0 + u_1x + u_2x^2 + \dots) + c_{t-1} x^{t-1} (u_0 + u_1x + u_2x^2 + \dots) - c_{t-1} u_0 x^{t-1} \\ &\quad + \dots + c_1 x (u_0 + u_1x + u_2x^2 + \dots) - (c_1 u_0 x + \dots + c_1 u_{t-2} x^{t-1}) + u_0 + u_1x + \dots + u_{t-1} x^{t-1} \\ &= c_t x^t f(x) + c_{t-1} x^{t-1} f(x) + \dots + c_1 x f(x) + g(x), \end{aligned}$$

where  $g(x) = (1 - c_t x^t - c_{t-1} x^{t-1} - \dots - c_1 x) f(x)$ . Since  $g(x)$  is a polynomial with a degree  $t$ ,

$$f(x) = \frac{g(x)}{(1 - c_t x^t - c_{t-1} x^{t-1} - \dots - c_1 x)}. \tag{5}$$

When the characteristic equation  $x^t = c_1 x^{t-1} + c_2 x^{t-2} + \dots + c_{t-1} x + c_t$  has distinct characteristic roots  $x_1, x_2, \dots, x_t$ ,  $1 - c_t x^t - c_{t-1} x^{t-1} - \dots - c_1 x = (1 - x_1 x)(1 - x_2 x) \dots (1 - x_t x)$ ,

$$f(x) = \frac{g(x)}{(1 - x_1 x)(1 - x_2 x) \dots (1 - x_t x)}. \tag{6}$$

From the partial fraction decomposition theorem, we have

$$f(x) = \frac{A_1}{1 - x_1 x} + \frac{A_2}{1 - x_2 x} + \dots + \frac{A_t}{1 - x_t x}, \tag{7}$$

where  $A_1, A_2, \dots, A_t$  are constants computed from  $c_1, c_2, \dots, c_t$ , and  $\frac{A_j}{1 - x_j x} = A_j \sum_{i=0}^{+\infty} x_j^i x^i$ . Then  $f(x) = \sum_{i=0}^{+\infty} (A_1 x_1^i + \dots + A_t x_t^i) x^i$ . Thus  $u_i = A_1 x_1^i + \dots + A_t x_t^i$ , which completes the proof.  $\square$

**Theorem 2.** Assume that  $u_i$  is an LFSR sequence having distinct characteristic roots  $x_1, x_2, \dots, x_l$  with respectively the multiplicities  $m_1, m_2, \dots, m_l$  such that  $m_1 + m_2 + \dots + m_l = t$ . Then

$$u_i = A_1(i) x_1^i + A_2(i) x_2^i + \dots + A_l(i) x_l^i, \tag{8}$$

where  $A_j(i) = P_1 + P_2 i + P_3 i^2 + \dots + P_{m_j} i^{m_j-1}$ ,  $j = 1, 2, \dots, l$ , and  $P_1, P_2, \dots, P_{m_j}$  are the undetermined coefficients that can be computed from  $c_1, c_2, \dots, c_t$ .

**Proof.** Let  $f(x) = u_0 + u_1x + \dots + u_{t-1}x^{t-1} + u_t x^t + \dots$ . From Theorem 1,  $f(x) = \frac{g(x)}{(1 - c_t x^t - c_{t-1} x^{t-1} - \dots - c_1 x)}$ . When the characteristic equation  $x^t = c_1 x^{t-1} + c_2 x^{t-2} + \dots + c_{t-1} x + c_t$  has repeated roots  $x_1, x_2, \dots, x_l$  with multiplicities  $m_1, m_2, \dots, m_l$  such that  $m_1 + m_2 + \dots + m_l = t$ ,  $(1 - c_t x^t - c_{t-1} x^{t-1} - \dots - c_1 x) = (1 - x_1 x)^{m_1} (1 - x_2 x)^{m_2} \dots (1 - x_l x)^{m_l}$ . Hence

$$f(x) = \frac{g(x)}{(1 - x_1 x)^{m_1} (1 - x_2 x)^{m_2} \dots (1 - x_l x)^{m_l}}. \tag{9}$$

From the partial fraction decomposition theorem, Eq. (9) can be decomposed as follows:

$$\begin{aligned} f(x) &= \frac{P_1^{(1)}}{1 - x_1 x} + \frac{P_1^{(2)}}{(1 - x_1 x)^2} + \dots + \frac{P_1^{(m_1)}}{(1 - x_1 x)^{m_1}} + \frac{P_2^{(1)}}{1 - x_2 x} + \frac{P_2^{(2)}}{(1 - x_2 x)^2} \\ &\quad + \dots + \frac{P_2^{(m_2)}}{(1 - x_2 x)^{m_2}} + \dots + \frac{P_l^{(1)}}{1 - x_l x} + \frac{P_l^{(2)}}{(1 - x_l x)^2} + \dots + \frac{P_l^{(m_l)}}{(1 - x_l x)^{m_l}}. \end{aligned}$$

Now we examine each term in the equation,

$$\begin{aligned} \frac{P_1^{(1)}}{1 - x_1 x} &= P_1^{(1)} \sum_{i=0}^{+\infty} (x_1 x)^i = \sum_{i=0}^{+\infty} P_1^{(1)} x_1^i x^i, \\ \frac{P_1^{(2)}}{(1 - x_1 x)^2} &= \frac{1}{x_1} \left[ \frac{P_1^{(2)}}{(1 - x_1 x)} \right]' = \frac{1}{x_1} P_1^{(2)} \left[ \sum_{i=0}^{+\infty} (x_1 x)^i \right] = P_1^{(2)} \left[ \sum_{i=0}^{+\infty} (i + 1) (x_1^i x^i) \right], \\ \dots &\quad \dots \\ \frac{P_1^{(m_1)}}{(1 - x_1 x)^{m_1}} &= \frac{d^{m_1-1} \left[ \frac{P_1^{(m_1)}}{(1 - x_1 x)} \right]}{dx^{m_1-1}} \cdot \frac{1}{x_1^{m_1-1} (m_1 - 1)!} = \frac{d^{m_1-1} \left[ \sum_{i=0}^{+\infty} x_1^i x^i \right]}{dx^{m_1-1}} \cdot \frac{P_1^{(m_1)}}{x_1^{m_1-1} (m_1 - 1)!} \\ &= \frac{P_1^{(m_1)} i(i-1) \dots (i - m_1 + 2)}{x_1^{m_1-1} (m_1 - 1)!} \sum_{i=m_1-1}^{+\infty} x_1^i x^{i+1-m_1} \\ &= \frac{P_1^{(m_1)} (i + m_1 - 1) \dots (i + 1)}{(m_1 - 1)!} \sum_{i=0}^{+\infty} x_1^i x^i. \end{aligned}$$

Then

$$f(x) = \sum_{i=0}^{+\infty} \left\{ \left[ P_1^{(1)} + P_1^{(2)}(i+1) + \dots + \frac{P_1^{(m_1)}(i+m_1-1)\dots(i+1)}{(m_1-1)!} \right] x_1^i + \left[ P_2^{(1)} + P_2^{(2)}(i+1) + \dots + \frac{P_2^{(m_2)}(i+m_2-1)\dots(i+1)}{(m_2-1)!} \right] x_2^i + \dots + \left[ P_l^{(1)} + P_l^{(2)}(i+1) + \dots + \frac{P_l^{(m_l)}(i+m_l-1)\dots(i+1)}{(m_l-1)!} \right] x_l^i \right\} x^i.$$

Thus

$$u_i = \left[ P_1^{(1)} + P_1^{(2)}(i+1) + \dots + \frac{P_1^{(m_1)}(i+m_1-1)\dots(i+1)}{(m_1-1)!} \right] x_1^i + \left[ P_2^{(1)} + P_2^{(2)}(i+1) + \dots + \frac{P_2^{(m_2)}(i+m_2-1)\dots(i+1)}{(m_2-1)!} \right] x_2^i + \dots + \left[ P_l^{(1)} + P_l^{(2)}(i+1) + \dots + \frac{P_l^{(m_l)}(i+m_l-1)\dots(i+1)}{(m_l-1)!} \right] x_l^i.$$

Therefore  $u_i = A_1(i)x_1^i + A_2(i)x_2^i + \dots + A_l(i)x_l^i$ , where  $A_j$  is a polynomial with degree  $m_j - 1$ . This completes the proof.  $\square$

### 2.2. The third-order LFSR sequence

Now we consider the third-order LFSR sequence [23,24]. Let  $f(x) = x^3 - ax^2 + bx - 1$ , where  $a, b \in F$ , be an irreducible polynomial over  $F = GF(p)$ . A sequence  $\mathbf{s} = \{s_k\}$  is said to be a third-order LFSR sequence with the characteristic polynomial  $f(x)$  if its elements satisfy

$$s_k = as_{k-1} - bs_{k-2} + s_{k-3}, \tag{10}$$

where  $s_0 = 3, s_1 = a$ , and  $s_2 = a^2 - 2b$ . We denote  $s_k$  as  $s_k(a, b)$  and  $\mathbf{s}$  as  $\mathbf{s}(a, b)$ , and call  $(a, b)$  the generator or base element of the LFSR sequence  $\mathbf{s}$  and  $k$  the exponent of  $s_k$ .

Assume that  $\alpha_1, \alpha_2, \alpha_3$  are the three roots of  $f(x) = 0$  over  $F$ . According to Newton’s formula, the elements of  $\mathbf{s}$  can be represented as follows by the symmetric  $k$ th-power sum of the roots:

$$s_k = \alpha_1^k + \alpha_2^k + \alpha_3^k, \quad k = 0, 1, \dots \tag{11}$$

Let us denote the period of  $f(x)$  by  $per(f)$ . Notice that if  $f(x)$  is irreducible over  $F$ , the period of  $\mathbf{s}(f)$  is equal to  $per(f)$ . The following results have been proved in [23,24].

**Lemma 3.** Let  $f(x) = x^3 - ax^2 + bx - 1$  be a polynomial over  $F = GF(p)$ ,  $\alpha_1, \alpha_2, \alpha_3$  be the three roots of  $f(x)$  in its extension field  $GF(p^3)$ , and  $\mathbf{s}$  be the characteristic sequence generated by  $f(x)$ . Let  $f_k(x) = (x - \alpha_1^k)(x - \alpha_2^k)(x - \alpha_3^k)$ , thus

1.  $f_k(x) = x^3 - s_k(a, b)x^2 + s_{-k}(a, b)x - 1$ , where  $s_{-k}(a, b) = s_k(b, a)$ .
2.  $f(x)$  and  $f_k(x)$  have the same period if and only if  $(per(f), k) = 1$ .
3. If  $(per(f), k) = 1, f(x)$  is irreducible over  $F$  if and only if  $f_k(x)$  is irreducible over  $F$ .

Let  $f^{-1}(x) = x^3 - bx^2 + ax - 1$ . Then  $f^{-1}(x)$  is the reciprocal polynomial of  $f(x)$  and  $s_{-k}(a, b)$  is the characteristic sequence over  $F$  generated by  $f^{-1}(x)$ . We call  $s_{-k}(a, b)$  the reciprocal sequence of  $s_k(a, b)$ .

**Theorem 4.** Commutative law: Let  $f(x) = x^3 - ax^2 + bx - 1$  be a polynomial over  $F$ , and  $\mathbf{s}$  be the characteristic sequence generated by  $f(x)$ . For all integers  $k$  and  $e$ , the  $k$ -th term of the characteristic sequence generated by the polynomial  $f_e$  equals the  $e$ -th term of the characteristic sequence generated by the polynomial  $f_k$ , which in turn equals the  $(ke)$ -th term of the characteristic sequence generated by the polynomial  $f(x)$ . In other words,

$$s_k(s_e(a, b), s_{-e}(a, b)) = s_{ke}(a, b) = s_e(s_k(a, b), s_{-k}(a, b)). \tag{12}$$

**Proof.** Assume that  $\alpha_1, \alpha_2, \alpha_3$  are the three roots of  $f(x)$  in the extension field  $GF(p^3)$ . Let  $\mathbf{s}$  be the characteristic sequence generated by  $f(x)$ . Then

$$\begin{aligned} f_e(x) &= (x - \alpha_1^e)(x - \alpha_2^e)(x - \alpha_3^e) \\ &= x^3 - (\alpha_1^e + \alpha_2^e + \alpha_3^e)x^2 + (\alpha_1^e\alpha_2^e\alpha_3^e) \left( \frac{1}{\alpha_1^e} + \frac{1}{\alpha_2^e} + \frac{1}{\alpha_3^e} \right) x - \alpha_1^e\alpha_2^e\alpha_3^e \\ &= x^3 - s_e(a, b)x^2 + s_{-e}(a, b)x - 1. \end{aligned}$$

Thus the  $k$ -th term of the characteristic sequence generated by the polynomial  $f_e(x)$  can be expressed by

$$s_k(s_e(a, b), s_e(a, b)) = (\alpha_1^e)^k + (\alpha_2^e)^k + (\alpha_3^e)^k = (\alpha_1^k)^e + (\alpha_2^k)^e + (\alpha_3^k)^e = s_{ek}(a, b).$$

This proves the theorem.  $\square$

**Fact 1 [23]:** Let  $k$  be a fixed positive integer. If  $k$  satisfies  $\gcd(k, p^i - 1) = 1$  for  $i = 1, 2, 3$ , then for  $\forall u, v \in F$ , the system of equations  $s_k(a, b) = u$  and  $s_{-k}(a, b) = v$  has a unique solution  $(a, b) \in F \times F$ . In other words,  $s_k(a, b)$  and  $s_{-k}(a, b)$  are orthogonal in  $F$  with respect to the variables  $a$  and  $b$ .

**Lemma 5 ([23]).** Let  $f(x) = x^3 - ax^2 + bx - 1$  be an irreducible polynomial of the period  $Q = p^2 + p + 1$  over  $F$  and  $\mathbf{s} = \{s_k\}$  be the characteristic sequence generated by  $f(x)$ . Let  $k$  and  $k'$  be different coset leaders modulo  $Q$ , with both  $k$  and  $k'$  being relatively prime to  $Q$ . Then

$$(s_k, s_{-k}) \neq (s_{k'}, s_{-k'}). \quad (13)$$

### 2.3. The LFSR-based public key cryptosystem

In this section, we illustrate the LFSR public key cryptosystem using the third-order characteristic sequences over  $Z_N$ . We choose two primes  $p$  and  $q$  such that  $N = pq$ . Then the period of the irreducible polynomial is  $\delta = (p^2 + p + 1)(q^2 + q + 1)$ . All computations here are performed in  $Z_N$ .

1. Public keys:  $e$  and  $N$ , such that  $\gcd(e, p^i - 1) = 1$  for  $i = 2, 3$ .
2. Private key:  $d$ , such that  $de = 1 \pmod{\delta}$ .
3. Enciphering: Given the plain text  $m = (m_1, m_2)$ , where  $0 < m_1, m_2 < N$ , the ciphertext  $c = (c_1, c_2)$  can be computed as  $c_1 = s_e(m_1, m_2)$  and  $c_2 = s_{-e}(m_1, m_2)$ .
4. Deciphering: Given the ciphertext  $c = (c_1, c_2)$  and the decryption key  $d$ , the plain text can be computed by  $m_1 = s_d(c_1, c_2)$  and  $m_2 = s_{-d}(c_1, c_2)$ .

Let  $PK_1$  and  $PK_2$  be two public-key cryptosystems that are based on the key spaces  $G_1$  and  $G_2$ , respectively. Denote by  $\#G_1$  ( $\#G_2$ ) the number of elements in  $G_1$  ( $G_2$ ). From the theory of cryptography, the public key and private key have the same key space in any public key cryptosystem. Assume that for the same security level we have  $\frac{\#G_1}{\#G_2} = r$ .

**Definition 1.** The **compression ratio** of the private key (public key) in  $PK_2$  with respect to that in  $PK_1$  is  $r$ .

**Theorem 6.** The compression ratio of the private key (public key) in an LFSR-based public key cryptosystem is 3 compared to a typical public key cryptosystem over  $GF(p)$ . This indicates that the number of bits for the private key (public key) of the LFSR-based public key cryptosystem is one-third of that in a system over  $GF(p)$  for the same security level.

**Proof.** The LFSR-based public key cryptosystem is based on the third-order LFSR sequence over  $GF(p)$ , which has the same security level as one over  $GF(p^3)$ . From Definition 1, the compression ratio of the private key (public key) in an LFSR-based public key cryptosystem is  $r = \frac{\#GF(p^3)}{\#GF(p)} = 3$  compared with any typical public key cryptosystem over  $GF(p)$ .  $\square$

### 2.4. Review of the YCH scheme

In this section, we briefly review a very popular  $(t, n)$ -threshold scheme, the YCH scheme [9].

Let  $S_1, S_2, \dots, S_k$  be the  $k$  secrets to be shared. Denote by  $f(r, w)$  any two-variable one-way function. The dealer randomly selects  $n$  secret shadows  $w_1, w_2, \dots, w_n$  and distributes  $w_i$  to participant  $M_i$  via a secure channel. Then the dealer randomly picks up an integer  $r$  and computes  $f(r, w_i)$  for  $i = 1, 2, \dots, n$ . Next, it performs the following steps.

If  $k \leq t$ .

1. Choose a large prime  $q$  satisfying  $S_i < q$  for  $i = 1, 2, \dots, k$ , select integers  $a_j$  such that  $0 < a_j < q$  for  $j = 1, 2, \dots, t - k$ , and then construct the following  $(t - 1)$ -th degree polynomial  $h(x) \pmod{q}$ :

$$h(x) = S_1 + S_2x + \dots + S_kx^{k-1} + a_1x^k + a_2x^{k+1} + \dots + a_{t-k}x^{t-1} \pmod{q}.$$

2. Compute  $Y_i = h(f(r, w_i)) \pmod{q}$  for  $i = 1, 2, \dots, n$ .
3. Publish  $(r, Y_1, Y_2, \dots, Y_n)$ .

If  $k > t$ .

1. Choose a large prime  $q$  satisfying  $S_1, S_2, \dots, S_k < q$ , then construct the following  $(k - 1)$ -th degree polynomial  $h(x) \pmod{q}$ :

$$h(x) = S_1 + S_2x + \dots + S_kx^{k-1} \pmod{q}.$$

2. Compute  $Y_i = h(f(r, w_i)) \pmod{q}$  for  $i = 1, 2, \dots, n$ .

3. Compute  $h(i) \bmod q$  for  $i = 1, 2, \dots, k - t$ .
4. Publish  $(r, Y_1, Y_2, \dots, Y_n, h(1), h(2), \dots, h(k - t))$ .

For a  $(t, n)$ -threshold scheme, we need the secret shares of at least  $t$  participants to recover the  $k$  secrets  $S_1, S_2, \dots, S_k$ . These participants pool their shares  $f(r, w_i)$ , based on which the polynomial  $h(x) \bmod q$  can be uniquely determined as follows:

If  $k \leq t$

$$\begin{aligned} h(x) &= \sum_{i=1}^t Y_i \prod_{j=1, j \neq i}^t \frac{x - f(r, w_j)}{f(r, w_i) - f(r, w_j)} \bmod q \\ &= S_1 + S_2x + \dots + S_kx^{k-1} + a_1x^k + a_2x^{k+1} + \dots + a_{t-k}x^{t-1} \bmod q \end{aligned}$$

else

$$\begin{aligned} h(x) &= \sum_{i=1}^t Y_i \prod_{j=1, j \neq i}^t \frac{x - f(r, w_j)}{f(r, w_i) - f(r, w_j)} \bmod q + \sum_{i=1}^{k-t} h(i) \prod_{j=1, j \neq i}^{k-t} \frac{x - j}{i - j} \bmod q \\ &= S_1 + S_2x + \dots + S_kx^{k-1} \bmod q. \end{aligned}$$

### 3. Verifiable multi-secret sharing scheme

In this section, we propose two new verifiable  $(t, n)$  multi-secret sharing schemes, denoted by Scheme-I and Scheme-II, based on the LFSR sequence, the three-order LFSR public key cryptosystem, and Shamir's scheme. In [23] the plain text is encrypted over the LFSR base element, while in our proposed schemes the plain text is encrypted over the exponent element.

#### 3.1. Scheme-I

##### 3.1.1. Initialization phase

We adopt the same notations as those used in the YCH scheme and denote by  $S_1, S_2, \dots, S_k$  the  $k$  secrets to be shared. Note that we do not require the communication channel between the dealer and each participant to be secure. First, the dealer chooses two strong primes  $p$  and  $q$  and computes  $N = pq$ . Next, it selects two positive integers  $a$  and  $b$  to get  $f(x) = x^3 - ax^2 + bx - 1$ , an irreducible polynomial over  $F = GF(p)$ . Finally the dealer publishes  $N, a$ , and  $b$ .

Each participant  $M_i$  randomly chooses an integer  $e_i$  from the interval  $[2, N]$  as its own secret shadow such that  $\gcd(e_i, p^k - 1) = 1$  for  $k = 2, 3$ , and computes  $(s_{e_i}(a, b), s_{-e_i}(a, b))$ . Then  $M_i$  provides  $(s_{e_i}(a, b), s_{-e_i}(a, b))$  and its identity number  $id_i$  to the dealer. For any pair of participants  $M_i$  and  $M_j$ , the dealer must ensure that  $(s_{e_i}(a, b), s_{-e_i}(a, b)) \neq (s_{e_j}(a, b), s_{-e_j}(a, b))$ . Each participant  $M_i$  publishes  $\{id_i, s_{e_i}(a, b)\}$ .

##### 3.1.2. Construction phase

The dealer selects an integer  $e_0$  from the interval  $[2, \delta]$  and computes  $d$  such that  $de_0 = 1 \bmod \delta$ , where  $\delta$  is the period of  $f(x) = x^3 - ax^2 + bx - 1$ . Then it performs the following steps:

1. compute  $(s_{e_0}(a, b), s_{-e_0}(a, b))$  and  $I_i = s_{e_0}(s_{e_i}(a, b), s_{-e_i}(a, b))$  for each participant  $M_i$ ;
2. publish  $\{s_{e_0}(a, b), s_{-e_0}(a, b), d\}$ ;
3. choose a hash function  $H$  and compute  $R_i = H(I_i)$  for each participant  $M_i$ ;
4. if  $k \leq t$ ,
  - (a) choose a prime  $q_1$  such that  $S_i < q_1, i = 1, 2, \dots, k$ ;
  - (b) select  $t - k$  integers  $a_j$  such that  $0 < a_j < q_1, j = 1, 2, \dots, t - k$ ;
  - (c) construct a  $(t - 1)$ -th degree polynomial  $h(x) = S_1 + S_2x + \dots + S_kx^{k-1} + a_1x^k + a_2x^{k+1} + \dots + a_{t-k}x^{t-1} \bmod q_1$ ;
  - (d) compute  $Y_i = h(R_i) \bmod q_1, i = 1, 2, \dots, n$ ;
  - (e) publish  $\{Y_1, Y_2, \dots, Y_n\}$ ;
- else
  - (a) choose a prime  $q_1$  such that  $S_i < q_1, i = 1, 2, \dots, k$ ;
  - (b) construct a  $(k - 1)$ -th degree polynomial  $h(x) = S_1 + S_2x + \dots + S_kx^{k-1} \bmod q_1$ ;
  - (c) compute  $Y_i = h(R_i) \bmod q_1$  and  $h(i) \bmod q_1$  for  $i = 1, 2, \dots, n$ ;
  - (d) publish  $\{Y_1, Y_2, \dots, Y_n, h(1), h(2), \dots, h(k - t)\}$ .

##### 3.1.3. Recovery and verification phase

Let  $\mathcal{M} = \{M_1, M_2, \dots, M_t\}$ . The members of  $\mathcal{M}$  will recover the secrets  $S_1, S_2, \dots, S_k$  based on the following procedure.

1. Each  $M_i$  computes  $I'_i = s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))$  to get the share, where  $e_i$  is the shadow of  $M_i$ .
2. The participant in  $\mathcal{M}$  verifies  $I'_i$  provided by  $M_i$ . If  $s_d(I'_i) = s_{e_i}(a, b)$ , then  $I'_i$  is true; otherwise  $I'_i$  is false, which means that  $M_i$  might be a cheater.

3. Each participant computes  $R'_i = H(I'_i)$ .
4. Recover the secrets: the polynomial  $h(x) \bmod q_1$  can be uniquely determined as follows:  
if  $k \leq t$

$$\begin{aligned} h(x) &= \sum_{i=1}^t Y_i \prod_{j=1, j \neq i}^t \frac{x - R'_j}{R'_i - R'_j} \bmod q_1 \\ &= S_1 + S_2x + \cdots + S_kx^{k-1} + a_1x^k + a_2x^{k+1} + \cdots + a_{t-k}x^{t-1} \bmod q_1 \end{aligned}$$

else

$$\begin{aligned} h(x) &= \sum_{i=1}^t Y_i \prod_{j=1, j \neq i}^t \frac{x - R'_j}{R'_i - R'_j} \bmod q_1 + \sum_{i=1}^{k-t} h(i) \prod_{j=1, j \neq i}^{k-t} \frac{x - j}{i - j} \bmod q_1 \\ &= S_1 + S_2x + \cdots + S_kx^{k-1} \bmod q_1. \end{aligned}$$

### 3.2. Scheme-II

Similarly to Scheme-I, Scheme-II does not require the establishment of a secure channel between each participant and the dealer. Let  $S_1, S_2, \dots, S_k$  be the  $k$  secrets to be shared. First, the dealer chooses two strong primes  $p$  and  $q$  and computes  $N = pq$ . Second, it selects two positive integers  $a$  and  $b$  to get  $f(x) = x^3 - ax^2 + bx - 1$ , an irreducible polynomial over  $F = GF(p)$ . Then the dealer picks up an integer  $\alpha \neq 0$  and computes  $c_1, c_2, \dots, c_t$  from the characteristic equation  $(x - \alpha)^t = x^t + c_1x^{t-1} + \cdots + c_t = 0$ . Next, it selects a different prime number  $q_1$  ( $q_1 < p < N$ ) such that  $q_1 > c_i, i = 1, 2, \dots, t$ . Finally, the dealer publishes  $\{N, a, b, \alpha, q_1\}$ .

Each participant  $M_i$  randomly chooses an integer  $e_i$  from the interval  $[2, N]$  as its own secret shadow such that  $\gcd(e_i, p^k - 1) = 1$  for  $k = 2, 3$  and computes  $(s_{e_i}(a, b), s_{-e_i}(a, b))$ . Then it provides  $(s_{e_i}(a, b), s_{-e_i}(a, b))$  and its identity number  $Id_i$  to the dealer. For each pair of participants  $M_i$  and  $M_j$ , the dealer must ensure that  $(s_{e_i}(a, b), s_{-e_i}(a, b)) \neq (s_{e_j}(a, b), s_{-e_j}(a, b))$ . Each  $M_i$  publishes  $\{id_i, s_{e_i}(a, b)\}$ .

#### 3.2.1. Construction phase

The dealer performs the following steps.

1. Randomly choose an integer  $e_0$  from the interval  $[2, \delta]$  and compute  $d$  such that  $de_0 = 1 \bmod \delta$ , where  $\delta$  is the period of  $f(x) = x^3 - ax^2 + bx - 1$ .
2. Compute  $(s_{e_0}(a, b), s_{-e_0}(a, b))$  and  $I_i = s_{e_0}(s_{e_i}(a, b), s_{-e_i}(a, b))$ .
3. Choose a hash function  $H$  and compute  $R_i = H(I_i)$  for each participant  $M_i$ .
4. Consider a homogeneous LFSR presented by the the following equation and compute  $u_i$  for  $t \leq i \leq n + k$ :

$$\begin{cases} u_0 = R_1, u_1 = R_2, \dots, u_{t-1} = R_t, \\ u_{j+t} + c_1u_{j+t-1} + \cdots + c_tu_j = 0 \bmod q_1 (j \geq 0). \end{cases} \quad (14)$$

5. Compute  $Y_i = R_i - u_{i-1}, t < i \leq n$  and  $r_i = S_i - u_{i+n}$  for  $1 \leq i \leq k$ .
6. Publish  $\{s_{e_0}(a, b), d, r_1, r_2, \dots, r_k, Y_{t+1}, Y_{t+2}, \dots, Y_n\}$ .

#### 3.2.2. Recovery and verification phase

Now we shall show how a participant can verify other participants' cheating actions and how  $t$  honest participants can recover the shared secrets.

1. Each participant  $M_i$  computes  $I'_i = s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))$  to get the share, where  $e_i$  is the shadow of  $M_i$ .
2. The participant can verify  $I'_i$  provided by  $M_i$ : if  $s_d(I'_i) = s_{e_i}(a, b)$ , then  $I'_i$  is true; otherwise  $I'_i$  is false, which means that  $M_i$  might be a cheater.
3. Suppose  $t$  arbitrary participants  $M_i$  pool their secret shares  $\{R_i\}$ . They could compute  $t$  terms of the homogeneous LFSR by their shares based on the following equation:

$$u_{i-1} = \begin{cases} R_i & \text{if } 1 \leq i \leq t, \\ R_i - Y_i & \text{if } t < i \leq n. \end{cases}$$

We could take one of the following two methods to recover the shared secrets.

**Method 1:** Choose Lagrange interpolation and  $t$  pairs  $(i - 1, u_{i-1}/\alpha^{i-1})$  to gain the  $(t - 1)$ -th degree polynomial  $P(x) \bmod q_1$  according to the formula

$$P(x) = \sum \frac{u_{i-1}}{\alpha^{i-1}} \prod_{j \neq i} \frac{x - j + 1}{i - j} \bmod q_1 = P_1 + P_2x^1 + \cdots + P_t x^{t-1} \bmod q_1.$$

From Theorem 2, we obtain  $u_j = P(j)\alpha^j \bmod q_1$  for all  $j \geq t$  and the shared secrets can be computed by  $S_i = u_{i+n} + r_i$  for  $1 \leq i \leq k$ .

**Method 2:** From Theorem 2, we can solve the following system of equations:

$$\begin{cases} u_0 = P_1 * \alpha^0 \\ u_1 = (P_1 + P_2) * \alpha \\ u_2 = (P_1 + P_2 * 2 + P_3 * 2^2) * \alpha^2 \\ \dots \\ u_{i-1} = (P_1 + P_2(i-1) + \dots + P_t(i-1)^{t-1})\alpha^{i-1} \bmod q_1 \end{cases} \quad (15)$$

to gain the unique solution  $P_1, P_2, \dots, P_t$ . Then we could obtain  $u_i = (P_1 + P_2i + \dots + P_t i^{t-1})\alpha^i \bmod q_1$  for  $\forall i \geq t$  and recover the shared secrets based on the formula  $S_i = u_{i+n} + r_i$  for  $1 \leq i \leq k$ .

### 3.3. Feasibility analysis

We shall analyze the feasibility of the share generation algorithm and the verification algorithm in Scheme-I and Scheme-II in this subsection.

1. In both schemes, it is absolutely impossible for the dealer to become a cheater. Because each participant selects its own shadow independently, we do not need a verification process between the dealer and the participants.
2. In both schemes, each participant  $M_i$  computes  $I'_i = s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b)) = s_{e_0}(s_{e_i}(a, b), s_{-e_i}(a, b)) = I_i$ . Other participants can use the published  $I'_i$ . Moreover, every participant can reuse his/her own shadow.
3. In verification, if a participant  $M_i$  does not have a cheating action,  $s_d(I'_i) = s_d(s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))) = s_{de_0}(s_{e_i}(a, b), s_{-e_i}(a, b)) = s_{e_i}(a, b)$  since  $de_0 = 1 \bmod \delta$ ; otherwise,  $M_i$  might be a cheater.

## 4. Security analysis

Our schemes are based on the security of Shamir's secret sharing scheme, the LFSR sequence, and the third-order LFSR public key cryptosystem. In this section, we analyze the security strengths of the proposed schemes by examining how they can counter several major possible attacks.

### 4.1. Security of Scheme-I

The security of Scheme-I is based on that of the third-order LFSR-based public key cryptosystem [23] and that of Shamir's secret sharing scheme [1]. Our analysis on the security of Scheme-I is presented below.

*Attack 1: Participant Cheating.*

*Analysis:* Assume that a certain participant, say  $M_i$ , intends to provide a wrong private key  $s_{e_j}$  to gain the secret(s). Thus  $M_i$  computes  $I'_i = s_{e_j}(s_{e_0}(a, b), s_{-e_0}(a, b))$  and broadcasts it. However, when receiving  $I'_i$  provided by  $M_i$ , other participants could verify the validity of  $I'_i$  by computing  $s_d(I'_i) = s_{e_j}(a, b) \neq s_{e_i}(a, b)$  because the  $Id_i$  and the  $s_{e_i}$  of  $M_i$  are published. Therefore it is easy to detect whether or not  $M_i$  provides an incorrect  $I'_i$ .

*Attack 2: Conspiracy attacks.*

*Analysis:* Assume that two participants  $M_i$  and  $M_j$  collude in order to recover the secrets. For example, they could exchange their  $s_{e_i}$  and  $s_{e_j}$  values. Thus  $M_i$  holds  $s_{e_j}$  and  $M_j$  holds  $s_{e_i}$ . Then  $M_i$  can compute  $s_d(I'_j) = s_{e_j}(a, b)$  and  $M_j$  can compute  $s_d(I'_i) = s_{e_i}(a, b)$ . Therefore  $M_i$  and  $M_j$  might be able to pass the verification. However, this is not true because all participants have published their  $Id$  and  $(Id, s_e)$  pairs, which means that the  $Id$  could not be tampered with. Thus other participants can easily recognize the conspiracy of the participants  $M_i$  and  $M_j$ .

*Attack 3: A plotter E may try to reconstruct the polynomial  $h(x) \bmod q_1$  when there are less than  $t$  participants available.*

*Analysis:* If plotter  $E$  wants to use fewer than  $t$  shares to reconstruct the polynomial  $h(x) \bmod q_1$ , the hardness is equivalent to the case that  $E$  successfully breaks Shamir's scheme, as Scheme-I is based on the security of Shamir's scheme.

*Attack 4: The plotter E tries to obtain the secret shadow  $e_i$  of the participant  $M_i$  from the public information  $I'_i$  and  $(s_{e_i}(a, b), s_{-e_i}(a, b))$ .*

*Analysis:* Assume that the plotter  $E$  intends to get the secret shadow  $e_i$  of the participant  $M_i$  from the public information  $I'_i$  and  $(s_{e_i}(a, b), s_{-e_i}(a, b))$ .  $E$  can compute  $I'_i = s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))$ , and form the polynomial  $f_{e_i} = x^3 - s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))x^2 + s_{-e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))x - 1$ . Since  $f_{e_0} = x^3 - s_{e_0}(a, b)x^2 + s_{-e_0}x - 1$  is irreducible,  $f_{e_i}$  is also irreducible according to Lemma 3. Assume that  $\alpha$  and  $\beta$  are the roots of  $f_{e_0}$  and  $f_{e_i}$ , respectively. Then  $\beta = \alpha^{e_i}$ . Once  $\alpha$  and  $\beta$  are known, deriving the exponent  $e_i$  is equivalent to computing the discrete logarithm in  $GF(p^3)$ . Solving the discrete logarithm in  $GF(p^3)$  is much harder than that in  $GF(p)$ . As we know that the discrete logarithm problem is NP-complete, it is secure to reuse the secret shadow.

### 4.2. Security of Scheme-II

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



**Table 1**

A comparison study on the security strengths of the five schemes.

Property	Type1 [21]	Type 2 [21]	Scheme in [22]	Scheme-I	Scheme-II
The dealer distributes the secret shadow	No	No	Yes	No	No
Verify the cheating action among the participants	Yes	Yes	Yes	Yes	Yes
Prevent $t - 1$ or fewer participants from recovering the secrets	Yes	Yes	Yes	Yes	Yes
Resist the conspiracy attack	No	No	No	Yes	Yes
No secure channel is needed	Yes	Yes	Yes	Yes	Yes
Prevent the participants from revealing each other's secret shadow after the recovery phase	Yes	Yes	Yes	Yes	Yes
Resist revelation of the secret shadow $s_j$ of the participant $M_j$ from the public information	Yes	Yes	Yes	Yes	Yes
Resist the cheating actions among participants	Yes	Yes	Yes	Yes	Yes

*Analysis:* There are two methods to recover the secrets. In Method 1,  $P(x)$  is based on the characteristic equation of the homogeneous LFSR. Assume that  $t - 1$  or fewer participants pool their secret shares. Then the participants could get  $t - 1$  or fewer pairs  $(i, u_i/\alpha^i)$  of  $P(x)$ . Therefore they have no way to reconstruct the  $(t - 1)$ -th degree polynomial  $P(x)$ , which means that they could not obtain any information regarding the secrets and the secret shadows. In Method 2, because there are at least  $t$  unknown quantities, the shared secrets and secret shares could not be gained by solving  $t - 1$  or less number of simultaneous equations when  $t - 1$  or fewer participants pool their secret shares.

*Attack 2:* Attackers want to reveal the secret shadow  $e_i$  of the participant  $M_i$  from the public information  $s_{e_i}$ .

*Analysis:* Assume that an attacker intends to get the secret shadow  $e_i$  of the participant  $M_i$  from the public information  $(s_{e_i}, s_{-e_i})$ . It can form the polynomial  $f_{e_i}(x) = x^3 - s_{e_i}x^2 + s_{-e_i}x - 1$  based on  $(s_{e_i}, s_{-e_i})$ . Because  $f(x) = x^3 - ax^2 + bx - 1$  is irreducible over  $F$ ,  $f_{e_i}$  is also irreducible according to Lemma 3. Assume that  $\alpha$  and  $\gamma$  are the roots of  $f$  and  $f_{e_i}$ , respectively. Then  $\beta = \gamma^{e_i}$ . Once  $\alpha$  and  $\gamma$  are known, deriving the exponent  $e_i$  is equivalent to computing the discrete logarithm, which is an NP-complete problem. Therefore it is secure to reuse the secret shadow, which means that the attacker could not reveal  $e_i$  from the public information.

*Attack 3:* The attacker tries to reveal the secret share  $I_i$  from the public information  $s_{e_0}$  and  $s_{e_i}$ .

*Analysis:* Because  $I_i = s_{e_0}(s_{e_i}(a, b), s_{-e_i}(a, b)) = s_{e_i}(s_{e_0}(a, b), s_{-e_0}(a, b))$ , the attacker must try to reveal  $e_i$  and  $e_0$  first. From Lemma 3, the security of  $e_0$  and  $e_i$  is based on the discrete logarithm problem. Thus it is impossible for the attacker to reveal  $I_i$  from  $s_{e_0}$  and  $s_{e_i}$ .

*Attack 4:* The participant  $M_i$  tries to reveal the secret shadow  $e_j$  of  $M_j$  from  $I_j$ , where  $j \neq i$ .

*Analysis:*  $I_j = s_{e_j}(s_{e_0}(a, b), s_{-e_0}(a, b))$ . According to Lemma 3, computing  $e_j$  from  $I_j$  needs solution of the discrete logarithm problem. Therefore it is absolutely impossible for any participant to get other participants' secret shadows.

To complete this section, we compare the security strengths of Scheme-I and Scheme-II with those of the two schemes (Type 1 and Type 2) in [21] and the scheme in [22] in terms of countering the resistance attack. The results are reported in Table 1.

## 5. Performance analysis

In this section, we discuss several important properties and analyze the performance of the proposed schemes.

### 5.1. Advantages of the proposed schemes

- Multiple secrets can be reconstructed simultaneously within a secret sharing session.
- Each qualified subset of participants is able to compute the shared secrets while the unqualified ones cannot obtain any information about the secrets.
- The knowledge of any  $t$  or more shares suffices to reconstruct the secrets and the knowledge of fewer than  $t$  shares is not enough to reconstruct the secrets; i.e. Scheme-I and Scheme II are  $(t, n)$ -threshold schemes.
- Each participant is allowed to check the validity of the shares of other participants and itself; i.e. our schemes are verifiable.
- Traditionally the dealer should also be verifiable as some/all of the participants may be prevented from reconstructing the original secrets due to the dishonesty of the dealer. Thus each participant is allowed to check whether the dealer is honest when distributing the shadows. However, in our schemes, the participants choose their own shadows, making it impossible for the dealer to cheat them. Therefore we do not need to verify the validity of the dealer.
- The shadow of each participant will never be disclosed in the recovery and verification phases and its reuse is secure; i.e. our schemes are multi-use schemes.

### 5.2. Computational complexity

It is obvious that the most time-consuming phases in our schemes are the verifiable phase and the recovery phase. In this subsection, we discuss the security and computational complexity of these two phases and compare our schemes with others.

**Table 2**  
A comparative analysis of the characteristics of our schemes and those in [21,22].

Property	Type1 [21]	Type2 [21]	Scheme in [22]	Scheme-I	Scheme-II
Having verification property	Yes	Yes	Yes	Yes	Yes
It is impossible for the dealer to cheat	Yes	Yes	Yes	Yes	Yes
Efficient recovery, (re)construction and verification	Yes	Yes	Yes	Yes	Yes
Security is based on different public key cryptosystem	RSA	RSA	RSA	LFSR	LFSR
The length of the private key in 1024-bit finite fields	1024 bits	1024 bits	1024 bits	340 bits	340 bits
The length of the public key in 1024-bit finite fields	1024 bits	1024 bits	1024 bits	340 bits	340 bits
Shadows are reusable when participants join/leave the group	Yes	Yes	Yes	Yes	Yes
The shadow is reusable when shared secrets are reconstructed	Yes	Yes	Yes	Yes	Yes
The dealer does not know the shadow of each participant	Yes	Yes	Yes	Yes	Yes
Time complexity of the recovery phase when $(k > t)$	$O(t^2)$	$O(t^2)$	$O(k^2)$	$O(k^2)$	$O(t^2)$

### 5.2.1. Verifiable phase

The two schemes (Type 1 and Type 2) in [21] are based on the RSA cryptosystem, with Type 1 requiring each participant to have one exponent while Type 2 needs  $n$  exponents. The RSA encryption takes  $O(N^3)$  unless small encryption exponents are used. The scheme in [22] is based on the discrete logarithm problem. Our schemes are based on the LFSR-based public key cryptosystem. Each participant  $M_i$  chooses  $e_i$  as its secret shadow according to the underlying cryptosystem, with  $e_i$  being the private key, and the public key  $s_{e_i}(a, b)$  being published. Note that the scheme in [22] and our schemes all require each participant to choose its own shadow, therefore it is impossible for the dealer to cheat. However, the key length in our schemes is much shorter. Considering a 1024-bit finite field at the same security level, the length of the private key can be represented by 340 bits in our schemes while the private key in [21,22] is three times longer.

### 5.2.2. Recovery phase

In our schemes, the recovery phase is the most time-consuming one. In Scheme-I, participants take the Lagrange interpolation polynomial to distribute secrets. The  $n$ -th degree polynomial can be reconstructed in time  $O(n^2)$  by using Lagrange interpolation. Therefore the recovery phase in Scheme-I can be reconstructed in time  $O(t^2)$  when  $t \geq k$  or  $O(k^2)$  when  $t < k$ . In Scheme-II, there exist two methods to recover the secrets, with the first one taking time  $O(t^2)$ . We can solve the  $t$  simultaneous equations in the second method in time  $O(t^2)$ , which is much faster than Scheme-I when  $k > t$ . Table 2 illustrates a comparative analysis of the characteristics of our schemes and those in [21,22].

### 5.3. Dynamic multi-secret sharing

In our proposed schemes, the participant  $M_i$ , the secret  $e_i$ , etc., can be dynamically operated. In this section, we discuss the scheme by considering a dynamic refresh, delete, and addition in accordance with practical settings.

1. When a new participant  $M_{new}$  joins the network, it selects its own shadow  $e_{new}$  and provides  $s_{e_{new}}(a, b)$  and  $id_{new}$  to the dealer. The dealer computes  $(s_{e_0}(a, b), s_{-e_0}(a, b))$  and  $I_{new} = s_{e_0}((s_{e_{new}}(a, b), s_{-e_{new}}(a, b)))$ . Then  $s_{e_{new}}(a, b)$  and  $id_{new}$  are published.

2. When we need to delete a participant  $M_{del}$ , the dealer only needs to erase  $s_{del}(a, b)$  and  $Id_{del}$ . If the deleted participant  $M_{del}$  wants to adopt its  $s_{del}$  to reconstruct the secret, it could not pass the verification.

## 6. Conclusions

Based on the LFSR sequence, we have proposed two new and efficient verifiable multi-secret sharing schemes in this paper. Our schemes have the same advantages compared to the previous ones, but have better performance and shorter private/public key length in comparison with those in [21,22] for the same security level. Analyses indicate that our schemes are computationally secure and efficient. Moreover, they are easy to implement and are applicable in practical settings.

Our future research lies in the following two directions. First, we intend to design a better VMSS, which has less computation and storage requirements but is better suited for practical situations. Second, we will investigate how to dynamically operate our schemes when a new secret is inserted or an old secret is deleted. This is a very important question with a lot of practical applications.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 60973114 and Grant 61170249, and the US National Science Foundation under grants (CNS-1017662, CNS-0963957), in part by Fundamental Research Funds for the Central Universities of China (No.CDJXS10182215), in part by the Natural Science Foundation project of CQCSTC under Grant 2009BA2024, and in part by the State Key Laboratory of Power Transmission Equipment & System Security and New Technology, Chongqing University, under Grant 2007DA10512711206.

## References

- [1] A. Shamir, How to share a secret, *Communications of the ACM* 22 (11) (1979) 612–613.
- [2] GR BLAKLEY, Safeguarding cryptographic keys, in: *Proc. NCC*, vol. 48, AFIPS Press, 1979, pp. 313–317.
- [3] H.F. Huang, C.C. Chang, A novel efficient  $(t, n)$  threshold proxy signature scheme, *Information Sciences* 176 (10) (2006) 1338–1349.
- [4] S. Iftene, General secret sharing based on the chinese remainder theorem with applications in e-voting, *Electronic Notes in Theoretical Computer Science* 186 (2007) 67–84.
- [5] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, in: *26th Annual Symposium on Foundations of Computer Science*, IEEE, 1985, pp. 383–395.
- [6] J. He, E. Dawson, Multistage secret sharing based on one-way function, *Electronics Letters* 30 (19) (1994) 1591–1592.
- [7] L. Harn, Efficient sharing (broadcasting) of multiple secrets, *IEE Proceedings Computers and Digital Techniques* 142 (1995) 237–240.
- [8] H.Y. Chien, J.K. Jan, Y.M. Tseng, A practical  $(t, n)$  multi-secret sharing scheme, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 83 (12) (2000) 2762–2765.
- [9] C.C. Yang, T.Y. Chang, M.S. Hwang, A  $(t, n)$  multi-secret sharing scheme\* 1, *Applied Mathematics and Computation* 151 (2) (2004) 483–490.
- [10] C.C. Chang, C.C. Lin, C.H. Lin, Y.H. Chen, A novel secret image sharing scheme in color images using small shadow images, *Information Sciences* 178 (11) (2008) 2433–2447.
- [11] D.S. Tsai, G. Horng, T.H. Chen, Y.T. Huang, A novel secret image sharing scheme for true-color images with size constraint, *Information Sciences* 179 (19) (2009) 3247–3254.
- [12] Y.F. Chen, Y.K. Chan, C.C. Huang, M.H. Tsai, Y.P. Chu, A multiple-level visual secret-sharing scheme without image size expansion, *Information Sciences* 177 (21) (2007) 4696–4710.
- [13] M.H. Dehkordi, S. Mashhadi, An efficient threshold verifiable multi-secret sharing, *Computer Standards & Interfaces* 30 (3) (2008) 187–190.
- [14] L. Chen, D. Gollmann, C. Mitchell, P. Wild, *Secret sharing with reusable polynomials*, in: *Information Security and Privacy*, Springer, 1997, pp. 183–193.
- [15] M. Liu, L. Xiao, Z. Zhang, Linear multi-secret sharing schemes based on multi-party computation, *Finite Fields and Their Applications* 12 (4) (2006) 704–713.
- [16] L.J. Pang, Y.M. Wang, A new  $(t, n)$  multi-secret sharing scheme based on shamir's secret sharing, *Applied Mathematics and Computation* 167 (2) (2005) 840–848.
- [17] C.W. Chan, C.C. Chang, A scheme for threshold multi-secret sharing, *Applied Mathematics and Computation* 166 (1) (2005) 1–14.
- [18] J. Shao, Z. Cao, A new efficient  $(t, n)$  verifiable multi-secret sharing (vmss) based on ych scheme, *Applied Mathematics and Computation* 168 (1) (2005) 135–140.
- [19] J. Zhao, J. Zhang, R. Zhao, A practical verifiable multi-secret sharing scheme, *Computer Standards & Interfaces* 29 (1) (2007) 138–141.
- [20] R.J. Hwang, C.C. Chang, An on-line secret sharing scheme for multi-secrets, *Computer Communications* 21 (13) (1998) 1170–1176.
- [21] M. Hadian Dehkordi, S. Mashhadi, New efficient and practical verifiable multi-secret sharing schemes, *Information Sciences* 178 (9) (2008) 2262–2274.
- [22] Z. Eslami, J. Zarepour Ahmadabadi, A verifiable multi-secret sharing scheme based on cellular automata, *Information Sciences* 180 (15) (2010) 2889–2894.
- [23] G. Gong, L. Harn, Public-key cryptosystems based on cubic finite field extensions, *IEEE Transactions on Information Theory* 45 (7) (1999) 2601–2605.
- [24] G. Gong, L. Harn, H. Wu, The GH public-key cryptosystem, in: *Selected Areas in Cryptography*, Springer, 2001, pp. 284–300.