The Journal of Systems and Software 85 (2012) 1325–1332

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

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

Efficient (n, t, n) secret sharing schemes

Yan-Xiao Liu [a], Lein Harn [b], Ching-Nung Yang [c,*], Yu-Qing Zhang [a,d]

[a] Key Lab of Computer Networks and Information Security of Ministry of Education, Xidian University, China
[b] CSEE Department, University of Missouri-Kansas City, USA
[c] Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
[d] National Computer Network Intrusion Protection Center, GUCAS, China

ARTICLE INFO

Article history:
Received 9 June 2011
Received in revised form 1 December 2011
Accepted 14 January 2012
Available online 30 January 2012

Keywords:
Secret sharing
Homomorphism
Multiple secrets
Verifiable secret sharing
t-Consistency

ABSTRACT

Recently, Harn and Lin introduced a notion of strong t-consistency of a (t, n) secret sharing scheme and proposed a strong (n, t, n) verifiable secret sharing (VSS). In this paper, we propose a strong (n, t, n) VSS which is more efficient than Harn and Lin's VSS. Using the same approach, we propose a (n, t, n) multi-secret sharing scheme (MSS) to allow shareholders to share $n - t + 1$ secrets. Also, the proposed (n, t, n) MSS can be modified to include the verifiable feature. All proposed schemes are unconditionally secure and are based on Shamir's (t, n) secret sharing scheme.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Secret sharing (SS) is one of main research topics in modern cryptography and has been studied extensively in the literature. Blakley (1979) and Shamir (1979) independently proposed SS solutions for safeguarding cryptographic keys. In a (t, n) SS, the dealer divides the secret into n shares and distributes shares to n shareholders in such a way that any t or more than t shares can reconstruct this secret; but any $t-1$ or fewer than $t-1$ shares cannot obtain any information of the secret.

There are vast research papers on this subject. Recently, Harn and Lin (2010) introduced a notion of strong t-consistency of a (t, n) SS and proposed a strong verifiable secret sharing (VSS) scheme. In this paper, we propose an efficient strong (n, t, n) VSS which is more efficient than Harn and Lin's VSS (Harn and Lin, 2010). In addition, we propose a (n, t, n) multi-secret sharing scheme (MSS) to allow shareholders to share $n - t + 1$ secrets. A verifiable (n, t, n) MSS (VMSS), which is based on the proposed (n, t, n) MSS, is also introduced. All proposed schemes are unconditionally secure and are simple variation of original Shamir's (t, n) SS scheme.

1.1. Related works

In Shamir's (t, n) SS scheme, it assumes that a mutually trusted dealer divides the secret into n shares and distributes each share to corresponding shareholder secretly. Chor et al. (1985) presented a notion of verifiable secret sharing (VSS). The property of verifiability enables shareholders to verify that their shares are consistent. VSS has become a fundamental tool in distributed cryptographic researches, including secure multiparty computation (MPC) (Beerliova and Hirt, 2008; Cramer et al., 2000) and Byzantine agreement (BA) protocol (Cachin et al., 2005). The security of VSSs can be classified into two types that are either computational security (Feldman, 1987) or unconditional security (Nikov and Nikova, 2005; Pedersen, 1992). For example, the security of Feldman's VSS (Feldman, 1987) is based on the hardness of solving the discrete logarithm, and the security of Pederson's VSS (Pedersen, 1992) and Nikov and Nikova's VSS (Nikov and Nikova, 2005) are unconditional security.

In Shamir's (t, n) SS (Shamir, 1979), a mutually trusted dealer is responsible to generate shares and distribute each share to corresponding shareholder secretly. Ingemarsson and Simmons (1991) introduced a new type of SS without the assistance of a mutually trusted dealer. The basic idea of this type of SS is that each shareholder also acts as a dealer to select a sub-secret and generate shares for other shareholders. The master secret is the summation of all sub-secrets. However, there is one potential problem in their design. That is, the number of shares kept by each shareholder is proportional to the number of shareholders in the scheme.

* Corresponding author at: Department of Computer Science and Information Engineering, National Dong Hwa University, #1, Sec. 2, Da Hsueh Rd., Hualien, Taiwan. Tel.: +886 3 8634025; fax: +886 3 8634010.
E-mail address: cnyang@mail.ndhu.edu.tw (C.-N. Yang).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2012.01.027

Therefore, the storage and management of shares are very complicated. Pedersen (1991) proposed a solution to overcome the previous mentioned problem. Harn and Lin (2010) denoted Pedersen's approach as a (n, t, n) SS in which the first parameter, n refers to the number of dealers, the second parameter, t refers to the threshold, and the third parameter, n refers to the number of shareholders, in the SS. Harn and Lin also introduced a new notion of strong t-consistency of shares and proposed a strong (n, t, n) VSS.

In the (t, n) SS, the secret is protected by multiple shareholders; however, it requires very large data expansion (i.e., t shares are needed to reclaim one secret). Therefore, the original Shamir's (t, n) SS is very inefficient as a conveyor of information (Blakley and Meddows, 1984). MSS, which allows multiple secrets to be shared among shareholders, is proposed to improve the efficiency of Shamir's (t, n) SS. There are various proposals of MSS. For example, MSSs proposed by Dehkordi and Mashhadi (2008), Shao and Cao (2005), Zhao et al. (2007) are based on polynomials. The security of these MSSs is based on some cryptographic assumptions and therefore, they are computationally secure.

1.2. Our contribution

In this paper, we propose a strong (n, t, n) VSS which is more efficient than Harn and Lin's strong (n, t, n) VSS (Harn and Lin, 2010). Following the same approach, we propose the efficient (n, t, n) MSS and (n, t, n) VMSS. All proposed algorithms are unconditionally secure. We summarize contributions of this paper below.

- An efficient strong (n, t, n) VSS which enables shareholders to verify their shares is proposed.
- An efficient (n, t, n) MSS which enables shareholders to share $n - t + 1$ secrets is proposed.
- An efficient (n, t, n) VMSS which enables shareholders to verify their shares and to share $n - t$ secrets is proposed.
- All proposed schemes are unconditionally secure.

1.3. Organization of the paper

In Section 2, we provide some preliminaries, including Shamir's (t, n) SS, the definitions of t-consistency and strong t-consistency, and the strong (n, t, n) VSS proposed by Harn and Lin (2010). In Section 3, we propose an efficient strong (n, t, n) VSS. Section 4 presents an efficient (n, t, n) MSS and the (n, t, n) VMSS. The conclusion is included in Section 5.

2. Preliminaries

Benaloh (1987) proposed a notion of t-consistency of a (t, n) SS. The property of t-consistency can be used to check the consistency of shares.

Definition 1. t-Consistency of shares (Benaloh, 1987). A set of n shares, $s_1, s_2, \ldots, s_n$ is t-consistent if any subset containing t shares defines the same secret.

Benaloh (1987) observed that shares, $s_1, s_2, \ldots, s_n$ in Shamir's (t, n) SS are t-consistent if and only if the interpolation of values $(1, s_1), (2, s_2), \ldots, (n, s_n)$ yields a polynomial of degree at most $t - 1$. However, Harn and Lin (2010) pointed out that the t-consistency of shares does not guarantee that shares satisfy the security requirement of a (t, n) SS. Assume that the dealer uses a polynomial having degree $t - 2$ to generate shares. Then, the shares are t-consistent; but the threshold of shares is $t - 1$. These shares violate the security requirements of a (t, n) SS since any $t - 1$ shares can obtain the secret. Harn and Lin (2010) proposed the definition of strong t-consistency to fix this security problem.

Definition 2. Strong t-consistency (Harn and Lin, 2010). A set of n shares, $s_1, s_2, \ldots, s_n$ is strong t-consistent if (a) any subset containing t or more than t shares defines the same secret; and (b) any $t - 1$ or fewer than $t - 1$ shares cannot define the same secret.

Definition 3. Strong VSS (Harn and Lin, 2010). A strong verifiable secret sharing scheme can verify that shares are strong t-consistent.

We observer that if shares in Shamir's (t, n) SS are generated by a polynomial having degree $t - 1$ exactly, shares are strong t-consistent and satisfy the security requirements of a (t, n) SS.

2.1. Previous schemes

Notations used in this paper and their descriptions are listed in Table 1. These notations will be used to describe schemes throughout this paper.

In the proposed schemes, we use the term sub-share generated from sub-polynomial of each dealer (shareholder), and the master share to represent the sum of sub-shares. Also, the terms "secret shares" and "verification shares" denote the shares are used for reconstructing the secret or verifying the strong t-consistency of secret shares.

2.1.1. Shamir's (t, n) SS (Shamir, 1979)

In Shamir's (t, n) SS, there are n shareholders $\{P_1, P_2, \ldots, P_n\}$, and a trusted dealer D. The dealer D randomly selects a polynomial $f(x)$ having degree $t - 1$, where the secret, $s = f(0)$. Then, the dealer generates n shares $(s_1, s_2, \ldots, s_n)$, by computing $s_i = f(i)$, for $i \in [1, n]$. The dealer sends each share to shareholder secretly. In secret reconstruction, any t shares, say $s_1, s_2, \ldots, s_t$, can reconstruct $f(x)$ following Lagrange interpolating formula as $f(x) = \sum_{j=1}^{t} f(j) \prod_{i=1, i \neq j}^{t} ((x - i)/(j - i))$. The secret is obtained as $s = f(0)$. We outline Shamir's (t, n) SS in Scheme 1.

2.1.2. The (n, t, n) SS in (Harn and Lin, 2010)

In the (n, t, n) SS (Harn and Lin, 2010), each shareholder also acts as a dealer. We assume that n dealers (shareholders) $\{P_1, P_2, \ldots, P_n\}$,

Secret selection phase:

The dealer D selects a random polynomial $f(x)$ having degree t–1 with the secret $s = f(0)$.

Shares generation phase:

The dealer generates n shares as $S_{t,n}(f(x)) = (s_1, s_2, \ldots, s_n)$. The dealer D sends each share s_i to shareholder P_i, secretly.

Secret reconstruction phase:

Any t shares, $(s_{i_1}, s_{i_2}, \ldots, s_{i_t})$, where $\{i_1, i_2, \ldots, i_t\} \subset \{1, 2, \ldots, n\}$, can reconstruct the secret as $R_{t,n}(s_{i_1}, s_{i_2}, \ldots, s_{i_t}) = f(x)$. The secret is $s = f(0)$.

Scheme 1. Shamir's (t, n) SS.

Table 1
Notations and their descriptions.

Notation	Description	Scheme[a]
s	A secret in the (t, n) SS, where $s \in GF(p)$ and p is a prime number.	
P_i	The ith shareholder P_i, $i \in [1, n]$.	
$S_{t,n}(\cdot)$	A dealer shares a secret, s by randomly choosing a polynomial $f(x)$ having degree $t-1$: $f(x) = \sum_{t=0}^{t-1} a_i x^i$ such that $s = a_0$ and a_i, $i \in [0, t-1]$, are in a finite field GF(p). Then, the dealer generates n shares $s_1, s_2, \ldots, s_n$ as $s_j = f(j)$, $j \in [1, n]$, where j is the ID of shareholder P_j. Finally, we denote $S_{t,n}(f(x)) = (s_1, s_2, \ldots, s_n)$	#1
$R_{t,n}(\cdot)$	Secret reconstruction function: any t shares (say$(s_1, s_2, \ldots, s_t)$) can be used for reconstructing $f(x)$ following Lagrange interpolating polynomial, where $f(x) = \sum_{j=1}^{t} s_j \prod_{i=1, i\neq j}^{t} (x-1)/(j-1)$. Finally, we denote $R_{t,n}(s_1, s_2, \ldots, s_t) = f(x)$, and the secret is $s = f(0)$. Note: this secret reconstruction using Lagrange interpolation is also used in other schemes.	
P_i	The ith shareholder P_i, $i \in [1, n]$. Also P_i acts as a dealer.	#2,
S_i	A sub-secret is independently chosen by each dealer P_i.	#3,
S	The master secret, $s = \sum_{i=1}^{n} S_i$.	#4,
$s_{i,j}$	Every P_i computes n secret sub-shares, which are generated from secret sub-polynomials. Generation of secret sub-shares is described as follows. A dealer (shareholder) P_i, $i \in [1, n]$, selects a secret sub-polynomial $f_i(x)$ of degree $t-1$ with his sub-secret $S_i = f_i(0)$ to generate n secret sub-shares $S_{t,n}(f_i(x)) = (s_{i,1}, s_{i,2}, \ldots, s_{i,n})$ (note: we herein use the name secret sub-polynomial to represent $f_i(x)$ since it is a polynomial of dealer P_i; also these secret sub-shares are used for generating secret master shares for reconstructing the master secret). Afterward, P_i sends $s_{i,j}$ to other P_j secretly, where $j \neq i$, At the end, every P_i will have secret sub-shares, $s_{j,i}$, for $j = 1, 2, \ldots, n$.	#5, #6
m_i	Each shareholder (dealer) P_i computes the secret master share from his n secret sub-shares, $m_i = \sum_{j=1}^{n} s_{j,i}$.	
$v_{i,j}^{l}$	Every P_i computes n verification sub-shares, which are generated from verification sub-polynomials. Generation of all verification sub-shares is described as follows. A dealer (shareholder) P_i, $i \in [1, n]$, selects k verification sub-polynomials $f_i^l(x)$, where $l = 1, 2, \ldots, k$, of degree $t-1$ to generate n verification sub-shares $S_{t,n}(f_i^l(x)) = (v_{i,1}^{l}, v_{i,2}^{l}, \ldots, v_{i,n}^{l})$ for each sub-polynomial. Afterward, P_i sends $v_{i,j}^{l}$ to other P_j secretly, where $j \neq i$. At the end, every P_i has $(k \times n)$ verification sub-shares, $v_{j,i}^{l}$, for $j = 1, 2, \ldots, n$, and $l = 1, 2, \ldots, k$.	#3
v_i^l	Each shareholder (dealer) P_i computes k verification master shares $V = \{v_i^l\}$, $l = 1, 2, \ldots, k$, where $v_i^l = \sum_{j=1}^{n} v_{j,i}^{l}$, from his verification sub-shares.	
v_i	All dealers (shareholders) $(P_1, P_2, \ldots, P_n)$ collaborate to select a set of weights $(w_1, w_2, \ldots, w_n)$, where $w_i (\neq 0) \in GF(p)$ and $(w_1, w_2, \ldots, w_n)$ should be linearly independent to $(1, 1, \ldots, 1)$. Then every P_i computes his verification master share $v_i = \sum_{j=1}^{n} w_j s_{j,i}$ (note: in the proposed strong (n, t, n) VSS, each P_i will receive $s_{j,i}$ (i.e., $f_j(i)$) from other dealers (shareholders) where $j \neq i$).	#4
M_l	Multi-secrets M_l, $l = 1, 2, \ldots, (n-t+1)$, shared in the proposed (n, t, n) MSS. All dealers (shareholders) $(P_1, P_2, \ldots, P_n)$ collaborate to select $n-t+1$ of n-tuple vectors, $\underline{e_1}, \underline{e_2}, \ldots, \underline{e_{n-t+1}}$, where all elements are in GF(p). Any selected vector $\underline{e_l} = (e_{l,1}, e_{l,2}, \ldots, e_{l,n})$, $1 \leq l \leq n$, should satisfy that any $(n-t+1)$-tuples in $\underline{e_l}$ is linearly independent to all the corresponding $(n-t+1)$-tuples in $\underline{e_r}$, $1 \leq r \leq n-t+1$, $r \neq l$. The lth secret is $M_l = \sum_{i=1}^{n} e_{l,i} f_i(0)$.	#5, #6
$m_{i,l}$	Each shareholder (dealer) P_i computes the master share for M_l from his n sub-shares, $m_{i,l} = \sum_{j=1}^{n} e_{l,j} s_{j,i}$.	

[a] #1: Shamir's (t, n) SS; #2: the (n, t, n) SS; #3: Harn and Lin's strong (n, t, n) VSS; #4: the proposed strong (n, t, n) VSS; #5: the proposed (n, t, n) MSS; #6: the proposed (n, t, n) VMSS.

participate in generating the master secret. We outline this scheme in Scheme 2.

2.1.3. Harn–Lin strong (n, t, n) VSS

Harn and Lin (2010) proposed a strong (n, t, n) VSS based on the (n, t, n) SS (Scheme 2). The (n, t, n) VSS is outlined in Scheme 3.

We note that if the sum of two polynomials has degree $t-1$ exactly, then either both polynomials have degree at most $t-1$ or both polynomials have degree larger than $t-1$. Since the degree of all remaining unrevealed polynomials has been verified to be $t-1$ exactly in Step (3), in Step (4) shareholders can conclude that the degree of polynomial $F(x) = \sum_{i=1}^{n} f_i(x)$ is at most $t-1$. Furthermore,

Master secret generation phase:

Each dealer P_i(shareholder) selects a random sub-polynomial $f_i(x)$ having degree t–1 and the sub-secret is $S_i = f_i(0)$. The master secret is $S = \sum_{i=1}^{n} S_i$.

Master shares generation phase:

(1) Each P_i computes sub-shares, $S_{t,n}(f_i(x)) = (s_{i,1}, s_{i,2}, \ldots, s_{i,n})$, following the same procedure as in Scheme 1.
(2) Each P_i sends $s_{i,j}$ to other P_j secretly, for j=1, 2, ..., n, and $j \neq i$.
(3) Each P_i computes the master share as $m_i = \sum_{j=1}^{n} s_{j,i}$ from n sub-shares, $s_{j,i}$, for j=1, 2, ..., n.

Master secret reconstruction phase:

Any t master shares, $(m_{i_1}, m_{i_2}, \ldots, m_{i_t})$, where $\{i_1, i_2, \ldots, i_t\} \subset \{1, 2, \ldots, n\}$, can reconstruct the interpolating polynomial as $R_{t,n}(m_{i_1}, m_{i_2}, \ldots, m_{i_t}) = f_1(x) + f_2(x) + \cdots + f_n(x) = F(x)$, following Lagrange interpolating formula and then obtains the master secret as $S = F(0) = \sum_{i=1}^{n} S_i$.

Scheme 2. (n, t, n) SS.

Master secret generation phase:

Same as in Scheme 2.

Master shares generation phase:

Same as in Scheme 2.

Verification phase:

(1) Each shareholder P_i selects k random verification sub-polynomials $f_i^l(x)$, having degree $t-1$ exactly, where l=1, 2, ..., k, and computes n verification sub-shares $S_{t,n}(f_i^l(x)) = (v_{i,1}^l, v_{i,2}^l, \ldots, v_{i,n}^l)$ for each verification sub-polynomial $f_i^l(x)$. P_i sends $v_{i,j}^l$ to other shareholder P_j secretly, for j=1, 2, ..., n, and $j \neq i$.

(2) Each P_i computes k verification master-shares, $v_i^l = \sum_{j=1}^n v_{j,i}^l$, using his sub-shares, $v_{j,i}^l$, for j=1, 2, ..., n, and l=1, 2, ..., k. At the end of this step, each P_i has k verification master shares, $V_i = \{v_i^l\}$, for l=1, 2, ..., k.

(3) All shareholders (P_1, P_2, ..., P_n) determine to reveal a subset U_i (say $|U_i|=k/2$) of V_i for verification. If the degree of all $k/2$ interpolating polynomials of the revealed verification master shares is $t-1$ exactly, the degree of interpolating polynomials of the remaining unrevealed verification master shares is also $t-1$ exactly with very high probability.

(4) Each shareholder P_i releases $k/2$ values of the additive sum of the master share and each remaining unrevealed verification master share. If the degree of all the $k/2$ interpolating polynomials of released values is $t-1$ exactly, shareholders can conclude that their master shares are generated by the polynomial $F(x)$ having $t-1$ exactly.

Master secret reconstruction phase:

Same as in Scheme 2.

Scheme 3. Harn and Lin's strong (n, t, n) VSS.

since $F(x)$ is the sum of all sub-polynomials selected by shareholders and each shareholder P_i has selected his sub-polynomial $f_i(x)$ with degree $t-1$ exactly, so the degree of $F(x) = \sum_{i=1}^n f_i(x)$ is $t-1$ exactly (i.e., shares are strong t-consistent). Furthermore, in this VSS, master shares are protected from the released additive sum of the master share and each unrevealed verification master share.

3. The proposed strong (n, t, n) VSS

In this paper, we propose an efficient strong (n, t, n) VSS, in which the sub-shares of the master secret are used for verification purpose. Notice that in Harn and Lin's scheme each shareholder needs to generate k additional verification sub-polynomials and sub-shares for the verification.

The homomorphism property of the following summation of polynomials

$$F(x) = f_1(x) + f_2(x) + \cdots + f_n(x) \tag{1}$$

has been used in the design of the (n, t, n) SS and strong (n, t, n) VSS (Harn and Lin, 2010). The homomorphism property implies that the additive sum of shares, $f_1(i)+f_2(i)+\cdots+f_n(i)$, of polynomials, $f_1(x), f_2(x), \ldots, f_n(x)$, is the share of additive sum of polynomials, $f_1(x)+f_2(x)+\cdots+f_n(x)$. The homomorphism property can also be applies to the following combination of polynomials

$$F_w(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x) \tag{2}$$

where $w_i \in GF(p)$ and $w_i \neq 0$, $1 \leq i \leq n$. It implies that the additive sum of shares, $w_1 f_1(i) + w_2 f_2(i) + \cdots + w_n f_n(i)$, of polynomials, $f_1(x)$, $f_2(x), \ldots, f_n(x)$, is the share of additive sum of polynomials, $w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x)$.

Our VSS uses the same master secret generation, master shares generation, and master secret reconstruction as in Scheme 2. Our VSS is outlined in Scheme 4.

Theorem 1. If the set of weights $(w_1, w_2, \ldots, w_n)$ is randomly selected and the degree of polynomial $F_w(x) = \sum_{i=1}^n w_i f_i(x)$ is

Master secret selection phase:

Same as in Scheme 2.

Master shares generation phase:

Same as in Scheme 2.

Verification phase:

(1) Shareholders, (P_1, P_2, ..., P_n), work together to select a set of weights, (w_1, w_2, ..., w_n), where $w_i (\neq 0) \in GF(p)$ and (w_1, w_2, ..., w_n) should be linearly independent to $(1, 1, \ldots, 1)$.

(2) Each P_i computes and publishes a verification share, $v_i = \sum_{j=1}^n w_j s_{j,i}$.

(3) All shareholders compute the interpolating polynomial on the n released verification shares, $R_{n,n}(v_1, v_2, \ldots, v_n) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x) = F_w(x)$. If the degree of the interpolating polynomial is $t-1$ exactly, n master shares are strong t-consistent (see the proofs in Theorems 1 and 2).

Master secret reconstruction phase:

Same as in Scheme 2.

Scheme 4. The proposed strong (n, t, n) VSS.

exactly $t-1$, then the degree of polynomial $F(x)=\sum_{i=1}^{n} f_i(x)$ is exactly $t-1$.

Proof. Suppose that there exists polynomials $f_i(x), 1 \le i \le n$, having degree larger than $t-1$ (say t). Let d_i be the coefficient associated with the term x^t of $f_i(x)$. Since the set of weights $(w_1, w_2, \ldots, w_n)$, is randomly selected by all shareholders, the probability that the degree of polynomial $F_w(x)=\sum_{i=1}^{n} w_i f_i(x)$ is $t-1$ exactly equals to the probability that $w_1 d_1 + w_1 d_2 + \cdots + w_n d_n = 0$. We observe that this probability is $(1/p)$ and this probability can be ignored if p is a large prime number. Thus, the degree of each polynomial $f_i(x)$, where $1 \le i \le n$, is at most $t-1$. In addition, let d_i' be the coefficient associated with the term x^{t-1} of $f_i(x)$, where $1 \le i \le n$, and d_w' be the coefficient associated with the term x^{t-1} of $F_w(x)$, we have $w_1 d_1' + w_2 d_2' + \cdots + w_n d_n' = d_w'$. Since $d_w' \neq 0$, we conclude that at least one of d_i', where $1 \le i \le n$, is nonzero. Let $d'(=\sum_{i=1}^{n} d_i')$ be the coefficient associated with the term x^{t-1} of $F(x)$. The probability that $d'=0$ is $(1/p)$ and this probability can be ignored (Harn and Lin, 2010). Thus, the degree of the polynomial $F(x)=\sum_{i=1}^{n} f_i(x)$ is exactly $t-1$. □

Theorem 2. The proposed efficient strong (n, t, n) VSS can verify that master shares are strong t-consistent without revealing the master secret and master shares.

Proof. Based on Theorem 1, if each shareholder can verify that the interpolating polynomial $F_w(x)$ of all the verification shares $v_i, 1 \le i \le n$, is of degree $t-1$ exactly, then shareholders can conclude that the degree of polynomial $F(x)=\sum_{i=1}^{n} f_i(x)$ is $t-1$ exactly. Since all the master shares, m_i, $1 \le i \le n$, are the shares of $F(x)$, all the master shares can be verified to be strong t-consistent. In addition, knowing the interpolating polynomial $F_w(x)$ in Step 3 cannot obtain the polynomial $F(x)$ of the master secret. Similarly, one cannot obtain any information about the master share, $m_i = s_{1,i} + s_{2,i} + \cdots + s_{n,i}$, from the corresponding verification share, $v_i = w_1 s_{1,i} + w_2 s_{2,i} + \cdots + w_n s_{n,i}$, since $(w_1, w_2, \ldots, w_n)$ is linear independent to $(1, 1, \ldots, 1)$. Thus, one cannot reveal the master share. □

Both our strong (n, t, n) VSS and Harn and Lin's strong (n, t, n) VSS verify the strong t-consistency property. However, the verification of Harn–Lin scheme is more complicate. Our strong (n, t, n) VSS is more suitable for real-environment application (e.g., applying VSS to electronic voting schemes) since the timeliness may be an important issue. Our scheme takes full advantage of secret sharing homomorphism to efficiently verify the strong t-consistency without the needing of additional k verification shares. Thus, our verification greatly improves speed and efficiency of verification procedure and, at the time, preserves the effectiveness of verification. On the other hand, in Harn and Lin's strong (n, t, n) VSS, if an attacker maliciously selects k verification polynomials, the probability that this attacker can compromise the VSS is $1/(k, k/2)$. But, in our proposed VSS, the probability that the attacker can compromise the VSS is $(1/p)$. If a prime number $p \approx 2^{128}$ is used for safeguarding a 128-bit secret key in AES, the parameter k in Harn and Lin's strong (n, t, n) VSS is about 132. This is the reason why in Harn and Lin's strong (n, t, n) VSS the parameter $k=100$ is chosen in their verification (note: the value of k will be up to 1030 for safeguarding a 1024-bit private key of RSA), and this value is incredibly too large for verification.

Both schemes (Harn and Lin's scheme and the proposed scheme) can verify the t-consistency of shares and are designed based on the homomorphism of shares. Both schemes are probabilistic VSS (see Step (3) in Scheme 3 and Theorem 2) with high probability to verify shares successfully. In general, the probabilistic VSS is more efficient than the deterministic VSS. In comparing with Harn and Lin's VSS, our scheme further enhances the efficiency to verify the strong t-consistency of master shares in using a weight vector instead of using k polynomials in Harn and Lin' scheme.

3.1. Security analysis

In our strong (n, t, n) VSS, all n shareholders are also dealers. In security analysis, we discuss whether colluded dealers (shareholders) can fail the verification. We also discuss whether colluded shareholders can reduce the threshold (i.e., t) of our scheme.

3.1.1. Colluded shareholders try to fail the verification

We consider the situation when c colluded shareholders, where $c<t$ and $t \le (n-c)$, try to fail the verification. The condition "$c<t$" assures that the colluded shareholders cannot reconstruct the secret by themselves, while "$t \le (n-c)$" assures that the honest shareholders can reconstruct the secret. Without loss of generality, suppose that c colluded shareholders are $\{P_1, P_2, \ldots, P_c\}$. We consider the case that colluded shareholders intentionally select secret sub-polynomials having degree not equal to $t-1$. There are two cases having degree not equal to $t-1$:

Case (1): selecting secret sub-polynomials having degree larger than $t-1$.

Suppose that $P_1, P_2, \ldots, P_c$ select their secret sub-polynomials $f_i(x), 1 \le i \le c$, having degree t. Let the coefficients associated with x^t of $f_i(x)$ be b_i. From Step (3) in the proposed strong (n, t, n) VSS, $F_w(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x)$, these b_is have to satisfy that $w_1 b_1 + w_2 b_2 + \cdots + w_c b_c = 0$ to fail our VSS successfully. Since the weight vector $(w_1, w_2, \ldots, w_n)$ is determined by all dealers (shareholders), the colluders are unable to guarantee that $w_1 b_1 + w_2 b_2 + \cdots + w_c b_c = 0$. Thus, our proposed scheme can detect this type of attack.

Case (2): selecting secret sub-polynomials having degree less than $t-1$.

Suppose that $P_1, P_2, \ldots, P_c$ select their sub-polynomials, $f_i(x)$, $1 \le i \le c$, having degree at most $t-2$. Since the master polynomial $F_w(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x)$ is the additive sum of all sub-polynomials selected by shareholders, the polynomial $F_w(x)$ is still having degree $t-1$ exactly. Thus, this type of attack cannot affect our scheme either.

3.1.2. Colluded shareholders try to reconstruct the secret by themselves

The scenario of second attack is that $t-1$ colluded shareholders (say $P_1, P_2, \ldots, P_{t-1}$) try to reconstruct the master secret by themselves based on the knowledge of (i) their sub-polynomials $(f_1(x), f_2(x), \ldots, f_{t-1}(x))$, or (ii) their sub-shares received from other shareholders $(s_{j,i} = f_j(i) t \le j \le n, 1 \le i \le t-1)$, or (iii) the verification master shares $(v_1, v_2, \ldots, v_n)$ or (iv) the set of weight vector $(w_1, w_2, \ldots, w_n)$.

Theorem 3. Any $t-1$ colluded shareholders cannot recover the master secret based on the knowledge of (i), (ii), (iii), or (iv).

Proof. The $t-1$ colluded shareholders only have their $t-1$ secret master-shaves. If we can prove that these $t-1$ colluders cannot gain the secret sub-polynomials of other shareholders, and thus they cannot have any other secret master-shaves. Therefore, they do not have enough master shares to recover the master secret.

Suppose the $t-1$ colluded shareholders (say $P_1, P_2, \ldots, P_{t-1}$) want to gain all the coefficients of $f_j(x) = f_{j,0} + f_{j,1} x + \cdots + f_{j,t-1} x^{t-1}$, which are the sub-polynomials selected by P_j, $t \le j \le n$, from

Multi secrets generation phase:

(1) Each shareholder P_i selects a random secret sub-polynomial $f_i(x)$ having degree t–1, with the sub-secret $S_i = f_i(0)$.

(2) All shareholders work together to select a set of n-tuple weight vectors, $\underline{e_i}$, for $i = 1, \ldots, n-t+1$, where $\underline{e_i} = (e_{i,1}, e_{i,2} \cdots, e_{i,n})$ should satisfy that any $(n-t+1)$-tuple elements in $\underline{e_l}$ $(l \in [1, n-t+1])$ are linearly independent to all the corresponding $(n-t+1)$-tuple elements in $\underline{e_r}$, for $r = 1, \ldots, n-t+1$, and $r \neq l$.

(3) The l-th master secret is $M_l = \sum_{i=1}^{n} e_{l,i} S_i$, l=1, 2, ..., $(n-t+1)$.

Master shares generation phase:

(1) Each P_i computes sub-shares $S_{t,n}(f_i(x)) = (s_{i,1}, s_{i,2}, \ldots, s_{i,n})$.

(2) Each P_i sends $s_{i,j}$ to other P_j secretly, where $j \neq i$.

(3) For each master secret M_l, P_i can compute the master share of the master secret from his n sub-shares as: $m_{i,l} = \sum_{j=1}^{n} e_{l,j}\, s_{j,i}$.

Master secrets reconstruction phase:

Any t shareholders release their master shares $\left(m_{i_1,l}, m_{i_2,l}, \ldots, m_{i_t,l}\right)$ to reconstruct the l-th secret M_l, $1 \leq l \leq n-t+1$, following Lagrange interpolating formula.

Scheme 5. The proposed (n, t, n) MSS.

(ii). The colluders have the following linear system with $(n-t+1)t$ unknowns (the coefficients $f_{j,i}$, $t \leq j \leq n$, $0 \leq i \leq t-1$) and $(n-t+1)(t-1)$ equations (for $1 \leq i \leq t-1$).

$$\begin{cases} f_t(i) = f_{t,0} + f_{t,1} i + \cdots + f_{t,t-1} i^{t-1} \\ f_{t+1}(i) = f_{t+1,0} + f_{t+1,1} i + \cdots + f_{t+1,t-1} i^{t-1} \\ \qquad\qquad \vdots \\ f_n(i) = f_{n,0} + f_{n,1} i + \cdots + f_{n,t-1} i^{t-1} \end{cases} \tag{3}$$

On the other hand, from (iii), the colluders can obtain $F_w(x) = \sum_{j=1}^{n} w_j f_j(x)$ through any t-out-of n public verification master shares v_i, $1 \leq i \leq n$. From (i), (iv) and $F_w(x)$, they can obtain $F'_w(x) = \sum_{j=t}^{n} w_j f_j(x)$. Suppose that C_i is the coefficient associated with x^i of $F'_w(x)$. Then, the colluders have:

$$\begin{cases} w_t f_{t,0} + w_{t+1} f_{t+1,0} + \cdots + w_n f_{n,0} = C_0 \\ w_t f_{t,1} + w_{t+1} f_{t+1,1} + \cdots + w_n f_{n,1} = C_1 \\ \qquad\qquad \vdots \\ w_t f_{t,t-1} + w_{t+1} f_{t+1,t-1} + \cdots + w_n f_{n,t-1} = C_{t-1} \end{cases} \tag{4}$$

Next, we prove that the number of linearly independent equations in combining Eq. (3) and Eq. (4) to solve $(n-t+1)t$ unknowns ($f_{j,i}$, $t \leq j \leq n$ and $0 \leq i \leq t-1$) is only $(n-t+1)(t-1)+1$. By Eq. (3), we can compute the following values:

$$\begin{cases} F'_w(1) = w_t f_t(1) + w_{t+1} f_{t+1}(1) + \cdots + w_n f_n(1) = A_1 \\ F'_w(2) = w_t f_t(2) + w_{t+1} f_{t+1}(2) + \cdots + w_n f_n(2) = A_2 \\ \qquad\qquad \vdots \\ F'_w(t-1) = w_t f_t(t-1) + w_{t+1} f_{t+1}(t-1) + \cdots + w_n f_n(t-1) \\ \qquad\qquad = A_{t-1} \end{cases} \tag{5}$$

Since $f_{j,0} = f_j(0)$, $t \leq j \leq n$, so the first equation in Eq. (4) can be rewritten as

$$F'_w(0) = w_t f_t(0) + w_{t+1} f_{t+1}(0) + \cdots + w_n f_n(0) = C_0 \tag{6}$$

From Eqs. (5) and (6), one can obtain $F'_w(x)$ using Lagrange interpolation without the help of Eq. (4). Therefore, the equations in Eq. (4) except the first equation (i.e., Eq. (6)) can be represented by Eq. (3) and Eq. (6). This implies that using Eqs. (3) and (4) to solve $(n-t+1) \times t$ unknowns in $f_j(x)$, $t \leq j \leq n$, the $t-1$ colluders only have $(n-t+1) \times (t-1) + 1$ linearly independent equations. So, $t-1$ colluders based on the knowledge of (i), (ii), (iii), or (iv) cannot obtain the sub-polynomials of other shareholders. □

4. The proposed (*n, t, n*) MSS

In this section, we propose a (n, t, n) MSS that allows shareholders to share $n-t+1$ secrets by using $n-t+1$ linearly independent sets of weight vector $(w_1, w_2, \ldots, w_n)$. The proposed (n, t, n) MSS is outlined in Scheme 5.

Theorem 4. The proposed (n, t, n) MSS shares up to $n-t+1$ linearly independent master secrets securely.

Proof. First, we want to show that the $n-t+1$ multiple master secrets shared in the proposed (n, t, n) MSS are linearly independent. In other words, we want to prove that any master secret in our proposed (n, t, n) MSS cannot be obtained by a linear combination of other $n-t$ master secrets. For each l, $l \in [1, n-t+1]$, the master secret M_l can be represented in a linear combination of all n sub-secrets with weights $(e_{l,1}, e_{l,2}, \ldots, e_{l,n})$, as $M_l = \sum_{i=1}^{n} e_{l,i} S_i$. Since any $(n-t+1)$-tuple vector in $\underline{e_l} = (e_{l,1}, e_{l,2}, \ldots, e_{l,n})$ is linearly independent to all corresponding $(n-t+1)$-tuple vectors in $\underline{e_r}$, for $r = 1, \ldots, n-t+1$, and $r \neq l$, it is easy to conclude that the weight vector, $\underline{e_1} = (e_{l,1}, e_{l,2}, \ldots, e_{l,n})$, of the master secret M_l is linearly independent to other $n-t$ weight vectors, $\underline{e_r}$, for $r = 1, \ldots, n-t+1$, $r \neq l$, of master secrets. Therefore, the master secret M_l cannot be obtained from the linear combination of other master secrets.

Second, we consider the situation that $t-1$ colluded shareholders (say $P_1, P_2, \ldots, P_{t1}$) want to recover the master secret by themselves. Here, we consider the worst case. When $n-t$ master secrets, $M_1, M_2, \ldots, M_{n-t}$ have already been reconstructed, these colluded shareholders want to recover M_{n-t+1} based on their own information and the revealed secrets, $M_1, M_2, \ldots, M_{n-t}$. In order to recover M_{n-t+1}, these colluded shareholders have to get an additional master share, $m_{k,n-t+1}$, $k \in [t, n]$ of M_{n-t+1}. Notice that the master share, $m_{k,n-t+1}$ can be represented as, $m_{k,n-t+1} = \sum_{j=1}^{n} e_{n-t+1,j} s_{j,k}$, where the sub-shares $s_{1,k}, s_{2,k}, \ldots, s_{t-1,k}$ are known to the colluded shareholders; but the remaining sub-shares, $s_{t,k}, s_{t+1,k}, \ldots, s_{n,k}$ are unknown to them. The first approach for, $P_1, P_2, \ldots, P_{t-1}$ to recover the master share, $m_{k,n-t+1}$ is to solve the sub-shares, $s_{t,k}, s_{t+1,k}, \ldots, s_{n,k}$ respectively. Let $m_{k,v}$, $v \in [1, n-t]$ be a master share of P_k which has been revealed. $m_{k,v}$ can be represented as $m_{k,v} = \sum_{j=1}^{n} e_{v,j} s_{j,k}$.

Multi secrets generation phase:
Same as Scheme 5.
/* the scheme shares n–t secrets M_i, $i = 2, 3, \ldots, n–t + 1$ */
Master shares generation phase:
Same as Scheme 5.
/* (1) the master shares of M_1 are used as the verification shares
(2) the master shares of $M_2 – M_{n-t+1}$ are used to reconstruct n–t secrets */
Verification phase:
Same as Scheme 4.
Master secrets reconstruction phase:
Same as Scheme 5.

Scheme 6. The proposed (n, t, n) VMSS.

Since $s_{1,k}, s_{2,k}, \ldots, s_{t-1,k}$ are known to the colluded shareholders, $m_{k,v}$ can be used to form an equation in terms of $s_{t,k}, s_{t+1,k}, \ldots, s_{n,k}$. Therefore, the number of available equations of, $s_{t,k}, s_{t+1,k}, \ldots, s_{n,k}$ are $n-t$ (i.e., $v \in [1, n-t]$). Since the number of unknowns $(n-t+1)$, is larger than the number of available equations $(n-t)$, the master share, $m_{k,n-t+1}$ cannot be solved in this approach. There is another approach to recover the master share, $m_{k,n-t+1}$. For the sake of simplicity, we can represent the master share, $m_{k,n-t+1}$, as $m_{k,n-t+1} = \sum_{j=1}^{t-1} e_{n-t+1,j} s_{j,k} + \sum_{j=t}^{n} e_{n-t+1,j} s_{j,k}$. If the colluded shareholders can compute, $m'_{k,n-t+1} = \sum_{j=t}^{n} e_{n-t+1,j} s_{j,k}$, they can solve the master share, $m_{k,n-t+1}$. However, the value $m'_{k,n-t+1} = \sum_{j=t}^{n} e_{nt+1,j} s_{j,k}$ cannot be computed in the linear combination of $n-t$ revealed values, $m'_{k,v} = \sum_{j=t}^{n} e_{v,j} s_{j,k}$, $v = 1, 2, \ldots, n-t$. This is because the set of n-tuple weight vectors, $\{\underline{e_1}, \underline{e_2}, \ldots, \underline{e_{n-t+1}}\}$, satisfies that any $(n-t+1)$-tuple vector in $\underline{e_{n-t+1}} = (e_{n-t+1,1}, e_{n-t+1,2}, \ldots, e_{n-t+1,n})$ is linearly independent to all corresponding $(n-t+1)$-tuple vectors in, $\{\underline{e_1}, \underline{e_2}, \ldots, \underline{e_{n-t}}\}$. Therefore, these $t-1$ colluded shareholders cannot get enough master shares to recover the master secret M_{n-t+1}. However, after $n-t+1$ master secrets and master shares being revealed, sub-shares of each shareholder can be determined. In conclusion, the proposed (n, t, n) MSS can only share $n-t+1$ different master secrets. □

In fact, our (n, t, n) MSS can be easily modified to include the feature of verifiable secret sharing. We need to trade one master secret for the verification of master shares. Let us use the first master shares $m_{i,1}$, for $i = 1, \ldots, n$, of master secret M_1 as the verification shares in our proposed strong (n, t, n) VSS (Scheme 4). Then, all shareholders compute the interpolating polynomial of the released verification shares. If the degree of the interpolating polynomial is $t-1$ exactly, any n master shares of master secret M_i, $i = 2, 3, \ldots, n-t+1$ are strong t-consistent. However, with the feature of verifiable secret sharing, the number of master secrets in (n, t, n) MSS is reduced by one. The proposed (n, t, n) VMSS is outlined in Scheme 6.

For the unanimous case (i.e., $t = n$), we can only share one secret in Scheme 5. On the other hand, the proposed (n, t, n) VMSS (Scheme 6) uses one secret for verification. Thus, there is no secret to be shared for the unanimous case. At this time, Schemes 5 and 6 are not MSS because Scheme 5 only shares one secret and Scheme 6 cannot share any secret. So the proposed (n, t, n) MSS and the proposed (n, t, n) VMSS can share two or more secrets for $t \le (n-1)$ and $t \le (n-2)$, respectively.

We now discuss the performance of the proposed (n, t, n) MSS and (n, t, n) VMSS, which has been addressed in MSS (Pang and Wang, 2005). (1) *Reusing of share in case of joining/leaving*: Each P_i in our MSS has two types of shares. One is the sub-share s_{ij}, $1 \le j \le n$, received from other shareholders, and the other one is master-share $m_{i,l}$, $1 \le l \le n-t+1$, determined from sub-shares. Therefore, if a new shareholder joins (say P_{n+1}), our scheme requires P_{n+1} to compute his sub-shares $s_{(n+1)j}$, $1 \le j \le n+1$, and to send them to other shareholders; also other shareholders need to compute $s_{i(n+1)}$, $1 \le i \le n$, and send them to P_{n+1}. If the shareholder (say P_n) leaves, it just needs to set the weighting $e_{l,n} = 0$. (2) *Reusing of sub-share for reconstructing multiple secrets*: In our MSS, sub-share s_{ij} can be reused, while master-share should be computed for each. (3) *Having verification property of shares*: Our VMS can verify the strong t-consistency property. (4) *Using one set of sub-shares per shareholder for reconstructing multiple secrets*: Each P_i in our (n, t, n) MSS has only one set of sub-shares s_{ij}, $1 \le j \le n$. (5) *Having no specific order for constructing multiple secrets*: In our MSS, multiple secrets can be reconstructed in any order. (6) *Having the same size of master share as the master secret*: The master share used for reconstructing the master secret has the same size as the master secret. (7) *Recovering multiple secrets in parallel*: Our MSS can reconstruct different secrets wither in a parallel way (revealing all secrets at one time) or in a serial way (revealing one secret at a time). If shareholders reveal all weights at one time, shareholders can reconstruct all secrets in parallel. On the other hand, if shareholders reveal their weights one at a time, shareholders can reconstruct the secret serially. (8) *Having no security assumption*: Our MSS and VMSS are unconditionally secure. However, most MSSs (Dehkordi and Mashhadi, 2008; Shao and Cao, 2005; Zhao et al., 2007) are based on the discrete logarithm assumption. (9) *No single dealer knowing of all shares*: In our MSS and VMSS, each shareholder acts as a dealer. The master secrets are determined by all shareholders. (10) *Capable to detect any cheating dealer*: Our VMSS can verify that all sub-shares are t-consistent. Thus, it can prevent the cheating caused by any malicious dealers (shareholders).

5. Conclusion

We propose a strong (n, t, n) VSS to verify the strong t-consistency of shares. This proposed scheme is more efficient than Harn and Lin's strong (n, t, n) VSS which was published most recently. In Harn and Lin's VSS, shareholders need to utilize 100 verification polynomials to verify the strong t-consistency of master shares. In our VSS, shareholders utilize the sub-polynomials of master secret to construct a verification polynomial and use them to verify master shares. In addition, we propose an efficient (n, t, n) MSS to allow shareholders to share $n-t+1$ secrets securely. The proposed (n, t, n) MSS is modified to become a (n, t, n) VMSS with verifiable feature. The security of all proposed schemes is unconditionally secure.

Acknowledgments

This work is supported in part by Testbed@TWISC, National Science Council under the Grant NSC 100-2219-E-006-001, the National Science Foundation of China under Grant Nos. 60970140,

60773135, 90718007, the National High-Tech Research and Development Program of China (863 Program) under Grant Nos. 2007AA01Z427, 2007AA01Z450.

References

Beerliova, Z., Hirt, M., 2008. Perfectly-secure MPC with linear communication complexity. In: Proceedings of the Fifth Theory of Cryptography Conference, vol. 4948, 19–21 March, New York, USA, LNCS. Springer-Verlag, Berlin, pp. 213–230.

Benaloh, J.C., 1987. Secret sharing homomorphisms: keeping shares of a secret. In: Advances in Cryptology, Proceedings of the Crypto'86, vol. 263, 11–15 August, Santa Barbara, California, USA, LNCS. Springer-Verlag, Berlin, pp. 251–260.

Blakley, G.R.,1979. Safeguarding cryptographic keys. In: Proceedings of the AFIPS'79 National Computer Conference 48. AFIPS Press, pp. 313–317.

Blakley, G.R., Meddows, C.,1984. Security of ramp schemes. In: Proceedings of CRYPTO'84. Springer-Verlag, Berlin, pp. 242–268.

Cachin, C., Kursawe, K., Shoup, V., 2005. Random oracles in Constantinople: practical asynchronous Byzantine agreement using cryptography. Journal of Cryptology 18 (3), 219–246.

Chor, B., Goldwasser, S., Micali, S., Awerbuch, B., 1985. Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, 21–23 October. IEEE Computer Society, Oregon, Portland, pp. 383–395.

Cramer, R., Damgard, I., Maurer, U., 2000. General secure multi-party computation from any linear secret sharing scheme. In: Advances in Cryptology, Proceedings of the Eurocrypt'00, vol. 1807, 14–18 May, Bruges, Belgium, LNCS. Springer-Verlag, Berlin, pp. 316–334.

Dehkordi, M.H., Mashhadi, S., 2008. New efficient and practical verifiable multi-secret sharing schemes. Information Sciences 178, 2262–2274.

Feldman, P., 1987. A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 27–29 October. IEEE Computer Society, Los Angeles, California, pp. 427–437.

Harn, L., Lin, C., 2010. Strong (n, t, n) verifiable secret sharing scheme. Information Sciences 180 (16), 3059–3064.

Ingemarsson, I., Simmons, G.J., 1991. A protocol to set up shared secret schemes without the assistance of a mutually trusted party. In: Advances in Cryptology, Proceedings of the Eurocrypt'90, vol. 473, 21–24 May, Aarhus, Denmark, LNCS. Springer-Verlag, Berlin, pp. 266–282.

Nikov, V., Nikova, S., 2005. On a Relation Between Verifiable Secret Sharing Schemes and a Class of Error-Correcting Schemes, Cryptology e-print archive 2003/210.

Pang, L.J., Wang, Y.M., 2005. A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing. Applied Mathematics and Computation 167, 840–848.

Pedersen, T.P., 1991. A threshold cryptosystem without a trusted party. In: Advances in Cryptology, Proceedings of the Eurocrypt'91, vol. 547, 8–11 April, Brighton, UK, LNCS. Springer-Verlag, Berlin, pp. 522–526.

Pedersen, T.P.,1992. Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances in Cryptology-CRYPTO'91, LNCS, vol. 576. Springer-Verlag, Berlin, pp. 129–140.

Shamir, A., 1979. How to share a secret. Communications of the ACM 22 (11), 612–613.

Shao, J., Cao, Z.F., 2005. A new efficient (t, n) verifiable multi-secret sharing (VMSS) based on YCH scheme. Applied Mathematics and Computation 168, 135–140.

Zhao, J., Zhang, J., Zhao, R., 2007. A practical verifiable multi-secret sharing scheme. Computer Standards and Interfaces 29 (1), 138–141.

Yan-Xiao Liu is currently a Ph.D. student in Xidian University, China. His current research interests include secret sharing and its applications.

Lein Harn received the B.S. degree in electrical engineering from the National Taiwan University in 1977, the M.S. degree in electrical engineering from the State University of New York-Stony Brook in 1980, and the Ph.D. degree in electrical engineering from the University of Minnesota in 1984. In 1984, he joined the Department of Electrical and Computer Engineering, University of Missouri- Columbia as an assistant professor, and in 1986, he moved to Computer Science and Telecommunication Program (CSTP), University of Missouri, Kansas City (UMKC). While at UMKC, he went on development leave to work in Racal Data Group, Florida for a year. His research interests include cryptography, network security, and wireless communication security. He has published a number of papers on digital signature design and applications and wireless and network security. He has written two books on security. He is currently investigating new ways of using secret sharing in various applications.

Ching-Nung Yang received the B.S. and M.S. degrees, both from the Department of Telecommunication Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1997. From 1987 to 1989, and from 1990 to 1999, he was with the Telecommunication Laboratory and with the Training Institute Kaohsiung Center, Chunghwa Telecom Company, Ltd., Kaohsiung, respectively. He is currently a Full Professor with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, and is also an IEEE senior member. He has published a number of journal and conference papers in the areas of information security and coding theory. He is the coeditor of the book "Visual Cryptography and Secret Image Sharing" published by CRC Press. His current research interests include coding theory and multimedia security.

Yu-Qing Zhang is a professor and supervisor of Ph.D. students of Graduate University of Chinese Academy of Sciences. He received his B.S. and M.S. degrees in computer science from Xidian University, China, in 1987 and 1990, respectively. He received his Ph.D. degree in Cryptography from Xidian University in 2000. His research interests include cryptography, wireless security, and trust management.