Wireless Pers Commun
DOI 10.1007/s11277-016-3862-z

(*t*, *n*) Multi-Secret Sharing Scheme Based on Bivariate Polynomial

Lein Harn[1] · **Ching-Fang Hsu**[1,2]

Abstract In a (*t*, *n*) secret sharing scheme (SS), a dealer divides the secret into *n* shares in such way that any *t* or more than *t* shares can reconstruct the secret but fewer than *t* shares cannot reconstruct the secret. The multi-SS is an extension of the (*t*, *n*) SS in which shares can be reused to reconstruct multiple secrets. Thus, the efficiency of the multi-SS is better than the efficiency of the (*t*, *n*) SS. In this paper, we propose the first multi-SS using a bivariate polynomial. Our design is unique in comparing with all existing multi-SSs. Shares generated using a bivariate polynomial can not only be used to reconstruct multiple secrets but also be used to establish pairwise keys between any pair of shareholders. The pairwise keys can protect exchange information in the secret reconstruction to prevent outsiders from obtaining the recovered secrets. All existing multi-SSs require additional key establishment to accomplish this.

Keywords Secret sharing scheme · Multiple secrets · Unconditional security · Bivariate polynomial

1 Introduction

The (*t*, *n*) secret sharing scheme (SS) was first introduced by Shamir [20] and Blakley [2] separately in 1979. In a (*t, n*) SS, a dealer divides a secret *s* into *n* shares and shared among a set of *n* shareholders such that (1) any *t* or more than *t* shares can reconstruct the secret *s*,

L. Harn and C.-F. Hsu have been contributed equally to this work.

✉ Ching-Fang Hsu
cherryjingfang@gmail.com

Lein Harn
harnl@umkc.edu

[1] Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, USA

[2] Computer School, Central China Normal University, Wuhan 430079, China

Published online: 31 October 2016

and (b) fewer than t shares cannot reconstruct the secret s. The (t, n) SS has become a fundamental tool in many applications such as in cloud computing [17, 18, 22] and in group communications [5, 6]. The (t, n) SS can be implemented using different mathematical tools. For example, Shamir's scheme [20] is based on a linear polynomial, Blakely's scheme [2] is based on the geometry, and Mignotte's scheme [16] and Asmuth-Bloom's scheme [1] are based on the Chinese remainder theorem (CRT).

In a (t, n) SS, during system set up, a dealer is responsible to divide a secret and generate n shares and sends each share to corresponding shareholder secretly. Later, shares are revealed by shareholders and used to reconstruct the secret. The efficiency of a (t, n) SS is very low since shares can only be used to reconstruct one secret. In order to improve its efficiency, multi-SS has been proposed in which shares generated by a dealer initially can be reused to reconstruct multiple secrets. We can classify all existing multi-SSs into two classes according to their security assumptions. The security assumption of the first class of schemes is based on a one-way function. For example, schemes in [9–11] are based on a one-way function and schemes in [15, 23] are based on a two-variable one-way function. The security assumption of the second class of schemes is based on some cryptographic assumptions, such as solving the discrete logarithm problem [8, 21] or the RSA assumption [14].

In all existing polynomial-based multi-SSs, shares are generated by univariate polynomials. Additional key establishment is needed for shareholders to protect their exchange information in the secret reconstruction such that the recovered secrets are not available to outsiders. In this paper, we propose the first multi-SS using a bivariate polynomial. One of unique features of our proposed scheme is that shares generated by an asymmetric polynomial can not only be used to reconstruct multiple secrets but also be used to establish pairwise keys between any pair of shareholders. The pairwise keys can protect exchange information in the secret reconstruction to prevent outsiders from obtaining the recovered secrets. Thus, no additional key establishment is needed. Recently, an unconditionally secure multi-SS [7] based on a univariate polynomial is proposed. But, each shareholder needs to keep multiple shares in [7]. In our proposed scheme, each shareholder only needs to keep one share.

Here, we summarize contributions of this paper.

- We propose the first multi-SS using an asymmetric bivariate polynomial.
- Each shareholder has only one share and the share can not only be used to reconstruct multiple secrets but also be used to establish pairwise keys to protect exchange information in secret reconstruction. Thus, outside attackers cannot obtain the recovered secrets.

The rest of paper is organized as follows. In Sect. 2, we review some preliminaries and then introduce the security model and design issues of our proposed multi-SS. Our proposed multi-SS based on a bivariate polynomial is presented in Sect. 3. The correctness, security analysis and performance and properties are included in Sect. 4. The conclusion is given in Sect. 5.

2 Preliminaries, Security Model and Design Issues

2.1 Secret Sharing Homomorphism

Benaloh [3] introduced the property of the secret sharing homomorphism. Let S be the domain of the secret and T be the domain of shares corresponding to the secret. The

function $F_I{:}T \rightarrow S$ is an induced function of the (t, n) SS. This function defines the secret s based on any subset containing t shares, $\{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$, such that $s = F_I(s_{i_1}, s_{i_2}, \ldots, s_{i_t})$, where $I = \{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$.

Definition 1: Homomorphism of the secret sharing [3] Let $\oplus$ and $\otimes$ be two functions on elements in sets S and T, respectively. We say that a (t, n) SS has the $(\oplus,\otimes)$-homomorphic property if for any subset I and $s = F_I(s_{i_1}, s_{i_2}, \ldots, s_{i_t})$, $s^{'} = F_I(s^{'}_{i_1}, s^{'}_{i_2}, \ldots, s^{'}_{i_t})$, then $s \oplus s^{'} = F_I(s_{i_1} \otimes s^{'}_{i_1}, s_{i_2} \otimes s^{'}_{i_2}, \ldots, s_{i_t} \otimes s^{'}_{i_t})$.

We note that shares generated by Shamir's (t, n) SS scheme satisfy $(+,+)$-homomorphism property. In other words, the sum of shares of two polynomials, $f(x)$ and $g(x)$, is the share of additive polynomial, $f(x) + g(x)$.

2.2 Bivariate Polynomial

A bivariate polynomial having degree $t - 1$ can be represented as $F(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \cdots + a_{t-1,t-1}x^{t-1}y^{t-1} \bmod p$, where $a_{i,j} \in GF(p)$, $\forall i, j \in [0, t - 1]$. A bivariate polynomial has been used in the design of a verifiable secret sharing scheme, we denote it as a BVSS. We can classify BVSSs into two types, the symmetric BVSSs, denote them as SBVSSs [4, 12, 13, 19] and the asymmetric BVSSs, denote them as ABVSSs, [4, 12]. If the coefficients of the polynomial satisfy $a_{i,j} = a_{j,i}$, $\forall i, j \in [0, t - 1]$, it is a symmetric bivariate polynomial. Shares generated by a bivariate polynomial can be used to establish pairwise keys between any pair of shareholders. In all (t, n) SBVSSs, a dealer selects a bivariate polynomial, $F(x, y)$, having degree $t - 1$ and $F(0, 0) = s$, where s is the secret. The dealer generates shares, $F(x_i, y) \bmod p$, $i = 1, 2, \ldots, n$, for shareholders, where p is a prime with $p > s$, and x_i is the public information associated with each shareholder, U_i. Each share, $F(x_i, y)$, is a univariate polynomial having degree $t - 1$. Since shares generated by a symmetric bivariate polynomial satisfy $F(x_i, x_j) = F(x_j, x_i)$, $\forall i, j \in [0, t - 1]$, pairwise keys, such as $F(x_i, x_j) = F(x_j, x_i)$, can be established between any pair of shareholders, U_i and U_j. On the other hand, in a ABVSS, the dealer generates a pair of shares, $F(x_i, y) \bmod p$ and $F(x, x_i) \bmod p$, $i = 1, 2, \ldots, n$, for each shareholder and the pairwise secret key, $F(x_i, x_j)$ or $F(x_j, x_i)$, can also be established between the pair of shareholders, U_i and U_j. For example, if we assume that the shared key between U_i and U_j is $F(x_i, x_j)$ for $x_i < x_j$. Then, the shareholder, U_i, can use his share, $F(x_i, y)$, to compute $F(x_i, x_j)$ and the shareholder, U_j, can use his share, $F(x, x_j)$, to compute $F(x_i, x_j)$.

2.3 Security Model

In our proposed secret reconstruction, each participated shareholder needs to use his share to compute and release a value to all other shareholders. The secrets can be recovered by each shareholder after receiving all exchange information from other shareholders. Secure channels are needed between any pair of shareholders in order to protect exchange information among shareholders; otherwise, outsiders can also obtain the recovered secret. A secure channel is referred to establish a shared secret key between two communication entities. Therefore, in a (t, n) SS, a key establishment protocol is also needed to protect the recovered secret from outsiders. All existing (t, n) SSs only address shares generation and secret reconstruction without including any key establishment scheme. In practical application, an additional key establishment is needed in the secret reconstruction process.

In this paper, we take a different approach to adopt a bivariate polynomial to generate shares by a dealer. Shares generated by a bivariate polynomial can not only be used to reconstruct multiple secrets but also be used to protect exchange information. Therefore, outsiders cannot obtain the recovered secrets. In our security model, we assume that attacks from outside adversaries are prevented since all exchange information in the secret reconstruction is protected by pairwise keys.

In the security analysis of our proposed scheme, we only consider attacks from inside adversaries. Inside adversaries is referred to be legitimate shareholders who own valid shares generated by the dealer. In a (t, n) SS, any $t - 1$ colluded shareholders should not be able to recover secrets. On the other hand, any t or more than t shareholders should be able to recover secrets. Furthermore, in the secret reconstruction of a multi-SS, the scheme needs (1) to protect the secrecy of shares; otherwise, shares cannot be reused for reconstructing multiple secrets; and (2) to protect the secrecy of uncovered secrets from knowing those recovered secrets.

2.4 Design Issues of a (t, n) Multi-secret Sharing Scheme

Here, we outline security issues in the secret reconstruction of a multi-SS.

(a) *Shares need to be protected in the secret reconstruction-* In a (t, n) multi-SS, multiple secrets are reconstructed one at a time. If shares are exposed to other shareholders in the secret reconstruction, shares cannot be reused to reconstruct uncovered secrets. In this paper, we propose to use Shamir's SS to protect the secrecy of shares. We let each shareholder to act like a dealer using Shamir's SS to generate sub-shares of his own share for all participated shareholders. Then, the additive sum of all sub-shares of each shareholder is released and used in the secret reconstruction. In this way, the secrecy of each share is protected since (1) it needs to compromise at least t sub-shares in order to obtain each share, and (2) it is impossible to derive each sub-share from the sum of sub-shares.

(b) *Recovered secrets should not affect the secrecy of uncovered secrets-* In a multi-SS, multiple secrets are reconstructed one at a time. Since some recovered secrets are already available to shareholders, our proposed scheme needs to prevent attackers from gaining additional information based on these recovered secrets to endanger the secrecy of uncovered secrets.

3 Our Proposed (t, n) Multi-SS

3.1 Scheme

In this section, we proposed a multi-SS using an asymmetric bivariate polynomial which can share multiple secrets, where t is the threshold, $h - 1$ is the degree of the variable y of the polynomial and k is the number of secrets.

3.1.1 Share Generation

The dealer selects an asymmetric polynomial, $F(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \cdots + \quad + a_{t-1,h-1}x^{t-1}y^{h-1} \bmod p$, where

$F(x, y)$ has $t-1$ degree in x and $h-1$ degree in y (i.e., $th > (t+h)(t-1)+(k-1)$ *and* $k \leq t$. We will explain this condition later in *Theorem 1*), $F(j, 0) = s_j$, $j = 1, 2, \ldots, k$, s_j, $j = 1, 2, \ldots, k$, are k secrets, $a_{i,j} \in GF(p)$, and p is a prime integer with $p > s_j$, $j = 1, 2, \ldots, k$. The dealer computes a pair of shares, a pair of shares, $s_i(y) = F(x_i, y)$ and $s_i(x) = F(x, x_i)$, for each shareholder, U_i, where $x_i > k$, $i = 1, 2, \ldots, n$, is the public information associated with each shareholder, U_i. The dealer sends each pair of shares, $(s_i(y), s_i(x))$, to shareholder U_i secretly.

3.1.2 Secret Reconstruction

Assume that u (i.e., $t \leq u \leq n$) shareholders, $\{U_{v_1}, U_{v_2}, \ldots, U_{v_u}\}$, want to reconstruct the secret, s_r, $r \in \{1, 2, \ldots, k\}$.

Step 1 Let us assume that the value, $F(x_i, x_j)$, with $x_{v_i} < x_{v_j}$, is used as the pairwise shared key between shareholders, U_i and U_j. Each shareholder U_{v_i} uses his/her one of shares, $s_{v_i}(y)$ or $s_{v_i}(x)$, to compute pairwise shared keys, $k_{i,j} = s_{v_i}(x_{v_j}) = F(x_{v_i}, x_{v_j})$ $j = 1, 2, \ldots, u$, $j \neq i$, where $k_{i,j}$ is the secret key shared between shareholders, U_{v_i} and U_{v_j}.

Step 2 Each shareholder U_{v_i} uses his share, $s_{v_i}(x)$, to compute the *Lagrange Component* (LC) of the secret, s_r, as $w_{v_i} = s_{v_i}(r) \prod_{l=1, l\neq i}^{u} \frac{r - x_{v_l}}{x_{v_i} - x_{v_l}} \bmod p$.

Step 3 Each shareholder U_{v_i} acts like a dealer to share his LC, w_{v_i}, using Shamir's (t, u)SS for all participated shareholders. In other words, each shareholder U_{v_i} selects a $t-1$ degree polynomial $f_i(x)$ with $f_i(0) = w_{v_i}$ and generates sub-shares, $w_{v_i,j} = f_i(x_{v_j})$, $j = 1, 2, \ldots, u$, for all participated shareholders.

Step 4 Each U_{v_i} computes $c_{i,j} = E_{k_{i,j}}(w_{v_i,j})$ $j = 1, 2, \ldots, u$, $j \neq i$, where $E_{k_{i,j}}(w_{v_i,j})$ denotes the conventional encryption of $w_{v_i,j}$ using the key $k_{i,j}$. Then, $c_{i,j}$, $j = 1, 2, \ldots, u$, $j \neq i$, are sent to other shareholders, respectively.

Step 5 After receiving ciphertext, $c_{j,i}$, $j = 1, 2, \ldots, u$, $j \neq i$, from other shareholders, each U_{v_i} computes $D_{k_{i,j}}(c_{j,i}) = w_{v_j,i}$, $j = 1, 2, \ldots, u$, $j \neq i$, where $D_{k_{i,j}}(c_{j,i})$ denotes the decryption of $c_{j,i}$using the key $k_{i,j}$.

Step 6 Each U_{v_i} computes $\sum_{j=1}^{u} w_{v_j,i} \bmod p = shr_i$ and computes $c_{i,j}^{'} = E_{k_{i,j}}(shr_i)$, $j = 1, 2, \ldots, u$, $j \neq i$, and sends $c_{i,j}^{'}$ to other shareholders, respectively.

Step 7 After receiving ciphertext, $c_{j,i}^{'}$, $j = 1, 2, \ldots, u$, $j \neq i$, from other shareholders, each U_{v_i} computes $D_{k_{i,j}}(c_{j,i}^{'}) = shr_j$, $j = 1, 2, \ldots, u$, $j \neq i$.

Step 8 Following Lagrange interpolation formula, the secret, s_r, can be computed as $\sum_{i=1}^{u} shr_i \prod_{l=1, l\neq i}^{u} \frac{-x_{v_l}}{x_{v_i} - x_{v_l}} \bmod p = s_r$.

3.2 Discussion

In Step 3, each shareholder uses Shamir's (t, u) SS to share his LG and generate sub-shares for all participated shareholders. In Steps 4 and 5, sub-shares are sent to other shareholders secretly using the pairwise shared keys computed from Step 1. In Step 6, each shareholder adds all received sub-shares to obtain the share corresponding to the additive sum of individual polynomials selected by participated shareholders. Note that the additive sum of constant terms of individual polynomials is the secret as shown in Step 8. In Steps 6 and 7,

the share of each shareholder is sent to other shareholders secretly using the pairwise shared keys. This prevent outsider from obtaining the recovered secret.

4 Analysis

4.1 Correctness

Since $F(x, 0)$ has $t-1$ degree in x and $F(r, 0) = s_r$, knowing any u (i.e., $t \le u \le n$) values, $F(x_{v_i}, 0)$, $i = 1, 2, \ldots, u$, the secret, s_r, can be obtained following the Lagrange interpolation formula as $s_r = F(r, 0) = \sum_{i=1}^{u} F(x_{v_i}, 0) \prod_{l=1, l\neq i}^{u} \frac{r - x_{v_l}}{x_{v_i} - x_{v_l}} \bmod p$. The *Lagrange Component* (LC) of the secret, s_r, of shareholder U_{v_i} is computed using the share, $s_r(x)$, in Step 2, as $w_{v_i} = s_{v_i}(r) \prod_{l=1, l\neq i}^{u} \frac{r - x_{v_l}}{x_{v_i} - x_{v_l}} \bmod p$. Note that $s_r = \sum_{i=1}^{u} w_{v_i}$. Each LC, w_{v_i}, cannot be released to other shareholders directly since the share value, $s_{v_i}(r)$, of shareholder U_{v_i} can be obtained from LC. In our scheme, Shamir's SS is adopted to protect the secrecy of shares. In Step 3, each shareholder U_{v_i} follows Shamir's (t, u) SS to select a $(t-1)$ degree polynomial $f_i(x)$ with $f_i(0) = w_{v_i}$ and generates sub-shares, $f_i(x_{v_j})$, $j = 1, 2, \ldots, u$, for all participated shareholders. In Steps 4 and 5, sub-shares are sent to other shareholder secretly using the pairwise shared keys computed from Step 1. In Step 5, each shareholder adds all received sub-shares to obtain the share corresponding to the additive sum of individual polynomials, $\sum_{i=1}^{u} f_i(x)$, selected by participated shareholders. Note that according to the *secret sharing homomorphism* [19], each additive sum of sub-shares is a share of the additive polynomial, $\sum_{i=1}^{u} f_i(x)$, with $\sum_{i=1}^{u} f_i(0) = \sum_{i=1}^{u} w_{v_i}$. Finally, in Step 8, according to the Lagrange interpolation formula, we can get $\sum_{i=1}^{u} shr_i \prod_{l=1, l\neq i}^{u} \frac{-x_{v_l}}{x_{v_i} - x_{v_l}} \bmod p = \sum_{i=1}^{u} f_i(0) = \sum_{i=1}^{u} w_{v_i} \bmod p = s_r$.

4.2 Security

As we have discussed in Sect. 2.2, in our scheme, we only need to consider attacks from inside adversaries. More specifically, we only consider attacks made by inside attackers. We analyze the security threat under the worst scenario. That is, there are at most $t-1$ colluded shareholders in the secret reconstruction.

Theorem 1 *With $th > (t + h)(t - 1) + (k - 1)$ and $k \le t$, the proposed scheme satisfies both security requirements of a (t, n) multi-SS. That are, (a) with t or more than t shares can recover the secrets, and (b) with fewer than t shares cannot recover the secrets.*

Proof $F(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \cdots + a_{t-1,h-1}x^{t-1}y^{h-1} \bmod p$, is an asymmetric polynomial having $t-1$ degree in x and $h-1$ degree in y, containing th different coefficients. In the proposed scheme, each pair of shares, $(s_i(y), s_i(x))$, is a univariate polynomial with degree $h-1$ in y and $t-1$ degree in x. In other words, each shareholder can use his share to establish $t + h$ linearly independent equations in terms of the coefficients of the polynomial $F(x, y)$. When there are $t-1$ colluded shareholders (i.e., the threshold is t) with their shares together, they can establish in total $(t + h)(t - 1)$ linearly independent equations. We want to point out that since in our proposed scheme, the LC of each shareholder is protected using Shamir's (t, u) SS, these collided shareholders cannot obtain LCs of other

shareholders. In addition, we consider the worst case in which the collided shareholders can gather the most information in the secret reconstruction process. That is, the $t - 1$ colluded shareholders want to recover the last secret after knowing $k - 1$ recovered secrets. So, the colluded shareholders can establish $k - 1$ additional equations of the coefficients of the polynomial $F(x, y)$ from these $k - 1$ previously reconstructed secrets. Thus, the total number of equations available to colluded shareholders is $(t + h)(t - 1) + (k - 1)$. The condition $th > (t + h)(t - 1) + (k - 1)$ ensures that the last secret cannot be recovered by $t - 1$ colluded shareholders. On the other hand, when there are t or more than t shareholders trying to recover the secret, with their shares together, they can establish $(t + h)t$ linearly independent equations. Since we have $(t + h)t \geq th$, any t or more than t shareholders can recover the secret.

Furthermore, since $F(x, 0)$ has $t - 1$ degree in x and k multiple secrets are defined as $F(i, 0) = s_y$, $i = 1, 2, \ldots, k$, after recovering t secrets, $F(x, 0)$is obtained. Thus, the number of secrets, k, needs to be limited as $k \leq t$. □

Corollary 1.1 *In our proposed scheme, the number of secrets k, the degree of the bivariate polynomial, F(x, y), and the threshold, t, have the following relation, $k \leq t$ and $h > t(t - 1) + k - 1$.*

Proof Form Theorem 1, since $th > (t + h)(t - 1) + (k - 1)$, we get $h > t(t - 1) + k - 1$. □

Let us examine two security issues in the secret reconstruction of a multi-SS as presented in Sect. 2.4.

(a) *Shares need to be protected in the secret reconstruction*- In Step 3 of our proposed scheme, each shareholder does not reveal his share to other shareholders directly. Instead, the share is hidden in a random polynomial and sub-shares are sent to other shareholders.

(b) *Recovered secrets should not affect the secrecy of uncovered secrets*- In the proof of Theorem 1, one additional equation of the coefficients of the polynomial $F(x, y)$can be established from each recovered secret. Our proof is based on the analysis of the worst case scenario in which the collided shareholders gather the most information in the secret reconstruction (i.e., by collecting $k - 1$ secrets which have been recovered already). Therefore, the proposed scheme allows multiple secrets to be recovered separately.

4.3 Performance and Properties of Our Scheme

4.3.1 Performance

Each shareholder needs to store a pair of shares, $(s_i(y), s_i(x))$, which are univariate polynomials having degree $h - 1$ and $t - 1$, respectively. In secret reconstruction, each shareholder needs to compute 2 LCs, in Steps 2 and 8 respectively, $u - 1$ pairwise shared keys, $k_{i,j}$, in Step 1, and $2(u - 1)$ conventional encryptions and decryptions in Steps 4, 6, 7. All these computations are much faster than most public-key computations since these computations involving multiplications/additions with a small modulus but public-key computations involving modular exponentiations with a large modulus. The communication complexity of this proposed scheme is $O(u)$, where u is the number of shareholders participated in the secret reconstruction.

4.3.2 Properties

(a) *No additional key establishment protocol is needed*- In our proposed scheme, shares generated by the dealer initially can also be used to establish pairwise shard keys between any pair of shareholders. There is no need to employ additional key establishment protocol. This property can speed up the secret sharing process significantly.
(b) *The proposed scheme can share up to k secrets*- As proved in Theorem 1, our proposed scheme is a unconditionally secure multi-secret sharing scheme. The shares can be reused repeatedly without compromising its security.

Remark 1 One major difference of our proposed (t, n) SS with most existing (t, n) SSs is that our scheme is based on a bivariate polynomial but all existing schemes are based on a univariate polynomial. The major advantage of using a bivariate polynomial is to enable shareholders to use their original shares obtained from the dealer initially to establish pairwise shared keys with other shareholders during secret reconstruction. Without pairwise shared keys to protect shares in the secret reconstruction, non-shareholder can also obtain shares and use them to recover the secret. Thus, our proposed scheme is more efficient than most existing schemes. However, in our proposed scheme, each share is a univariate polynomial; but in most other schemes, each share is an integer. Since computational complexity of our proposed scheme is $O(u)$, where u is the number of participated shareholders in secret reconstruction, the computational complexity of our scheme is the same as most existing schemes.

5 Conclusion

We propose the first (t, n) multi-SS using a bivariate polynomial. Our proposed scheme has two unique features in comparing with most existing multi-SSs. That are, (1) shares generated by the dealer can be used to reconstruct multiple secrets and (2) to protect exchange information in the secret reconstruction. We also include a detailed discussion to prove the correctness and security of our proposed scheme.

References

1. Asmuth, C. A., & Bloom, J. (1983). A modular approach to key safeguarding. *IEEE Transactions on Information Theory, 30*(2), 208–210.
2. Blakley, G. R. (1979). Safeguarding cryptographic keys. In *Proceedings of AFIPS'79 national computer conference* (Vol. 48, pp. 313–317). Montvale: AFIPS Press.
3. Benaloh, J. C. (1987). Secret sharing homomorphisms: Keeping shares of a secret. In *Advances in cryptology—CRYPTO'86, in LNCS* (Vol. 263, pp. 251–260). New York: Springer.
4. Gennaro, R., Ishai, Y., Kushilevitz, E., & Rabin, T. (2001). The round complexity of verifiable secret sharing and secure multicast. In *STOC* (pp. 580–589).
5. Harn, L., & Lin, C. (2010). Authenticated group key transfer protocol based on secret sharing. *IEEE Transactions on Computers, 59*(6), 842–846.
6. Harn, L. (2013). Group authentication. *IEEE Transactions on Computers, 62*(9), 1893–1898.
7. Harn, L. (2014). Secure secret reconstruction and multi-secret sharing schemes with unconditional security. *Security and Communication Networks, 7*(3), 567–573.
8. Harn, L. (1995). Efficient sharing (broadcasting) of multiple secrets. *IEE Computers and Digital Techniques, 142*(3), 237–240.

9. Harn, L. (1995). Comment multistage secret sharing based on one-way function. *Electronic Letters, 31*(4), 262.
10. He, J., & Dawson, E. (1994). Multistage secret sharing based on one-way function. *Electronic Letters, 30*(19), 1591–1592.
11. He, J., & Dawson, E. (1995). Multi-secret sharing scheme based on one-way function. *Electronic Letters, 31*(2), 93–94.
12. Katz, J., Koo, C., & Kumaresan, R. (2008). Improved the round complexity of VSS in point-to-point networks. In *Proceedings of ICALP '08, Part II, in: LNCS* (Vol. 5126, pp. 499–510). New York: Springer.
13. Kumaresan, R., Patra, A., & Rangan, C. P. (2010). The round complexity of verifiable secret sharing: the statistical case. In *Advances in cryptology—ASIACRYPT 2010, in: LNCS* (Vol. 6477, pp. 431–447). New York: Springer.
14. Lin, T. Y., & Wu, T. C. (1999). (*t*, *n*) threshold verifiable multisecret sharing scheme based on factorisation intractability and discrete logarithm modulo a composite problems. *IEE Proceedings of Computers & Digital Techniques, 146*(5), 264–268.
15. Lin, H. Y., & Yeh, Y. S. (2008). Dynamic multi-secret sharing scheme. *International Journal of Contemporary Mathematical Sciences, 3*(1), 37–42.
16. Mignotte, M. (1983). How to share a secret. In *Cryptography-proceedings of the workshop on cryptography, lecture notes in computer science* (Vol. 149, pp. 371–375). New York: Springer.
17. Nirmala, S. J., Bhanu, S. M. S., & Patel, A. A. (2012). A comparative study of the secret sharing algorithms for secure data in the cloud. *International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2*(4), 63–71.
18. Nojoumian, M., & Stinson, D. R. (2012). Social secret sharing in cloud computing using a new trust function. In *Proceeding of 2012 tenth annual international conference on privacy, security and trust* (pp. 16–167).
19. Nikov, V., & Nikova, S. (2005). On proactive secret sharing schemes. In *LNCS* (Vol. 3357, pp. 308–325). New York: Springer.
20. Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612–613.
21. Shao, J., & Cao, Z. (2005). A new efficient (*t*, *n*) verifiable multi-secret sharing (VMSS) based on YCH scheme. *Applied Mathematics and Computation, 168*(1), 135–140.
22. Takahashi, S., & Iwamura, K. (2013). Secret sharing scheme suitable for cloud computing. In *Proceeding of 2013 IEEE 27th international conference on advanced information networking and applications* (pp. 530–537).
23. Yang, C. C., Chang, T. Y., & Hwang, M. S. (2004). A (*t*, *n*) multi secret sharing scheme. *Applied Mathematics and Computation, 151*, 483–490.

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. He is currently a Professor at the Department of Electrical and Computer Engineering, University of Missouri, Kansas City (UMKC). He is currently investigating new ways of using secret sharing in various applications.

Ching-Fang Hsu received the M.Eng. and the Ph.D. degrees in information security from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2010 respectively. From Sep. 2010 to Mar. 2013, she was a Research Fellow at the Huazhong University of Science and Technology. She is currently an Assistant Professor at Central China Normal University, Wuhan, China. Her research interests are in cryptography and network security, especially in secret sharing and its applications.