Noname manuscript No.
(will be inserted by the editor)

An Efficient Multistage Secret Sharing Scheme Using Linear One-way Functions and Bilinear Maps

Mitra Fatemi · Taraneh Eghlidos · Mohammadreza Aref

the date of receipt and acceptance should be inserted later

Abstract In a Multistage Secret Sharing (MSSS) scheme, the authorized subsets of participants could reconstruct a number of secrets in consecutive stages. A One-Stage Multisecret Sharing (OSMSS) scheme is a special case of MSSS schemes that all secrets are recovered simultaneously. In these schemes, in addition to the individual shares, the dealer should provide the participants with a number of public values related to the secrets. The less the number of public values, the more efficient the scheme. It is desired that MSSS and OSMSS schemes provide the computational security; however, we show in this paper that OSMSS schemes do not fulfill the promise. Furthermore, by introducing a new multi-use MSSS scheme based on linear one-way functions, we show that the previous schemes can be improved in the number of public values. Compared to the previous MSSS schemes, the proposed scheme has less complexity in the process of share distribution. Finally, using bilinear maps, the participants are provided with the ability of verifying the released shares from other participants. To the best of our knowledge, this is the first verifiable MSSS scheme in which the number of public values linearly depends on the number of the participants and the secrets and which does not require secure communication channels.

Keywords Bilinear map · Linear one-way function · Multistage secret sharing · Multisecret sharing · Verifiability

M. Fatemi and M. Aref
Dept. of Electrical Engineering, Sharif University of Technology, Tehran, Iran
E-mail: {mitfatemi, aref}@sharif.edu

T. Eghlidos
Electronics Research Institute, Sharif University of Technology, Tehran, Iran
E-mail: teghlidos@sharif.edu

1 Introduction

In many applications, some piece of private information is required. For example, the parties involved in a secure communication may need to agree upon a secret to safeguard the transmitted data against misuse. However, secrets are always exposed to unauthorized access or getting lost. To prevent a secret from the above mentioned risks, one could benefit from a secret sharing scheme.

In a secret sharing scheme, a secret is shared among a set of parties, called participants, such that only authorized subsets of them could recover the secret. To make it happen, usually a trusted third party (dealer) assigns a confidential value to each participant as his/her share (shadow). Having these values, authorized subsets of participants could compute the secret through a prespecified rule. The set, including all of the authorized subsets, is called an access structure. An access structure could be chosen arbitrarily or it could be based on a threshold rule. In a (t, n)-threshold secret sharing scheme, each of the n participants is given a private share, such that every subset of at least t participants could recover the secret from their shares.

The concept of secret sharing was first introduced by Shamir [1] and Blakley [2] in 1979, independently. Many other secret sharing schemes were presented since then, for example [3] and [4]. Some various features were considered as well such as verifiability of the shares [5], [6], resistance against the presence of a number of cheaters [7], [8], generalized access structures [4] and dynamic change of the threshold and the number of participants [9], [10].

A (t, n)-threshold secret sharing scheme is called perfect, if less than t participants neither could reconstruct the secret, nor obtain any information about it

[11]. Note that here, the unconditional security is considered, that is, no limitation is presumed on the computational power of the participants. It has been shown that in a perfect secret sharing scheme, the size of the shares should not be less than the size of the secret [11]; in the case of equality, the scheme is referred to as ideal [12]. However, it is possible to decrease the share size below the secret size, if it suffices to have the computational security [13]. It should be notified that as stated in [13], "computational security is in no way a practical limitation. In fact, most implementations of theoretically perfect secret sharing schemes result in actual computational security".

A multistage secret sharing scheme (MSSS) is a generalization of a secret sharing scheme, where there are more than one secret to be shared. Still, each participant receives one master-shadow, the size of which is the same as the size of each secret. Thus, MSSS schemes could only provide computational security. The first MSSS scheme was proposed by He and Dawson in 1994 [14] and further improved in [15], [16], [17], [18], [19], [20], [21]. In MSSS schemes, the secrets could be recovered in different stages, possibly according to a prespecified order. In these schemes, there are some public values in addition to the master-shadows. The participants, who wish to participate in a secret reconstruction stage, derive the corresponding sub-shadows from their master-shadows and the public values.

In 2000, a one-stage multisecret sharing (OSMSS) scheme was proposed by Chien et al. [22]. In such a scheme, by assigning each of the participants a private shadow, a number of secrets are shared among them. However, when an authorized subset of participants pull their shares together, they recover all the secrets simultaneously in one stage, hence the name. Numerous OSMSS schemes were proposed thereafter to reduce the computational complexity and the number of public values [23], [24], [25], [26], [27], [28].

In this paper, we briefly study some of the previous MSSS and OSMSS schemes. The results of this investigation show that the least number of public values published so far is equal to $m \times (n-t)$ in the former schemes [16] and equal to $m+n-t+1$ in the latter schemes [24]. In the case of verifiable OSMSS scheme, the least number of public values increases to $2n+1$ [29] and $2(n+1)+m-t$ [28], where m, n and t indicate the number of secrets, the number of participants and the threshold value, correspondingly. We also show that the OSMSS scheme has an inherent security problem, that is, it cannot provide the desired level of computational security. More specifically, every unauthorized subset of $t-k$ participants are able to reduce the size of the secret space from $|\mathcal{S}|^m$ to $|\mathcal{S}|^k$, where $\mathcal{S}$ indicates the secret space and $0 < k < \min(m,t)$. Because of such a security dilemma in OSMSS schemes, we only focus on MSSS schemes, thereafter. Here, we propose an MSSS scheme, using a linear one-way function [30], which reduces the number of public values from $m \times (n-t)$ to m. Note that this is just equal to the number of secrets and is independent of the number of participants.

Next, we propose a modified version of the new scheme to acquire share verifiability with $2(n+1)+m-t$ public values. To the best of our knowledge, this is the first published verifiable MSSS scheme in which the participants could verify the correctness of other shares in the reconstruction stage. The proposed scheme also does not require any secure channel for transmitting the share values.

The rest of this paper is organized as follows. In section 2, we review the main concepts of secret sharing schemes. In section 3, we briefly study some of the previous MSSS and OSMSS schemes. The security flaw of OSMSS schemes is introduced in section 3 and then proved in section 4. The new MSSS scheme and its verifiable version are presented in section 5. A thorough analysis of the new schemes along with a comparison with the previous schemes are given in the subsequent section. We conclude the paper in section 7.

2 Definitions

Let $P=\{P_1,P_2,...,P_n\}$ be the set of n participants and $\mathcal{S}$ and $\mathcal{S}_P$ be the sets of all possible values for the secret (the secret space) and the shares (share space), respectively. A secret sharing scheme is a method that guarantees accessibility and security of a secret $s \in \mathcal{S}$, simultaneously. In such a scheme, each participant $P_i, 1 \leq i \leq n$ receives a private share $s_{P_i} \in \mathcal{S}_P$ such that only prespecified subsets of the participants, called authorized subsets, are able to recover the secret, using their shares. The set of all the authorized subsets is named an access structure, denoted by Γ. A threshold secret sharing scheme is a specific but crucial type of secret sharing with an access structure of the form

$$\Gamma = \{A \subseteq P : |A| \geq t\} \tag{1}$$

where t is the threshold value.

A (t,n)-threshold secret sharing scheme with a secret $s \in \mathcal{S}$ and the shares $s_1, s_2, ..., s_n \in \mathcal{S}_P$ is called perfect, if for every set of t indices $\{i_1,...,i_t\} \subset \{1,...,n\}$, the two following conditions hold, where $H(.)$ is the entropy function.

$$H(s|s_{i_1},...,s_{i_{t-1}},s_{i_t}) = 0$$
$$H(s|s_{i_1},...,s_{i_{t-1}}) = H(s) \tag{2}$$

In [11], it has been shown that the above constraints demand

$$H(s_i) \geq H(s), i = 1, .., n \tag{3}$$

Assuming that the secret and the shares are uniformly distributed in $\mathcal{S}$ and $\mathcal{S}_P$, respectively, which is usually the case, the relation (3) yields $|\mathcal{S}_p| \geq |\mathcal{S}|$, where $|A|$ represents the size of the set A. The scheme in which $|\mathcal{S}_p| = |\mathcal{S}|$ is called an ideal secret sharing scheme [12]. In other words, in an ideal secret sharing scheme, all the shares have the same size as the secret. Being ideal is a desired feature since the transmission and storage of large shares are not profitable. Moreover, large share size brings about redundancy in the scheme that reduces the efficiency.

A secret sharing scheme is composed of share distribution and secret reconstruction processes. The share distribution process is usually accomplished by a third authorized participant, called dealer. The dealer calculates the shares and sends them to the participants through secure channels. In the secret reconstruction process, an authorized set of participants cooperate to recover the secret.

A dilemma of the secret sharing schemes is that they are not resistant against the cheating participants who provide false share values to prevent the others from recovering the correct secret. This problem could be solved by adding verifiability of the shares to the scheme. In a verifiable secret sharing scheme, usually the dealer publishes some extra values during the share distribution process. By the help of these values, each participant is able to verify the correctness of the shares that are provided by other participants, before recovering the secret. The first verifiable secret sharing schemes were presented by Chor et al. [5] and Stadler [6], in 1985 and 1996, respectively. Since then, numerous verifiable secret sharing schemes have been presented, for example [31] and [32]. The former scheme uses the points on an elliptic curve and the discrete logarithm problem to achieve verifiability, while the latter utilizes a bilinear map.

Perfect secret sharing schemes have the two following limitations:

1. In each scheme just one secret is shared among the participants; and
2. After the secret recovery, all of the shares, even the shares of those participants who have not joined the secret reconstruction process, get revealed.

Hence, to share another secret among these participants, the dealer has to send them a new set of shares. Multistage secret sharing (MSSS) schemes and one-stage multisecret sharing (OSMSS) schemes have been presented as efficient solutions to overcome the mentioned constraints of the secret sharing schemes. In an MSSS scheme, a number of secrets are shared among the participants by assigning just one share (a master share) to each. All master shares have the same size as each of the secrets. Moreover, the participants are able to reconstruct the secrets in different stages, without jeopardizing the security of uncovered secrets. An OSMSS scheme is a special case of MSSS schemes, in which all the secrets get revealed simultaneously. It is noteworthy that based on the lower bound on the size of the shares in a perfect secret sharing scheme, given in equation (3), both MSSS and OSMSS schemes could not provide perfect security. Hence, we are just looking for the computational security in these schemes.

3 Review of previous MSSS and OSMSS schemes

In this section, we briefly review some of the previous MSSS and OSMSS schemes. The pioneer MSSS scheme presented by He and Dawson in 1994 [14] is based on the Shamir's secret sharing scheme and exploits the concept of public shift values. Let $f : \mathbb{Z}_p \to \mathbb{Z}_p$ be an arbitrary one-way function. For every $x \in \mathbb{Z}_p$ and every positive integer k, $f^k(x)$ denotes the result of k successive applications of f on x, that is, $f^k(x) = f\big(f^{k-1}(x)\big)$ and $f^0(x) = x$. Let $x_1, x_2, ..., x_n$ be n label values corresponding to the n participants. To share the secrets $S_1, ..., S_m$ among the participants $P_1, ..., P_n$ according to a (t, n)-threshold scheme, the dealer performs the following steps:

1. Chooses n arbitrary integers $s_1, ..., s_n$.
2. For each secret $S_i, i \in \{1, ..., m\}$, chooses random values $a_{i_1}, a_{i_2}, ..., a_{i_{t-1}} \in \mathbb{Z}_p$ and generates the polynomial
$$Q_i(x) = S_i + a_{i_1}x + ... + a_{i_{t-1}}x^{t-1} \tag{4}$$
Then, he computes the shift values as $d_{i,j} = Q_i(x_j) - f^{i-1}(s_j), j \in \{1, ..., n\}$.
3. Sends the master-shares $s_1, s_2, ..., s_n$ through secure channels to the participants $P_1, P_2, .., P_n$, respectively and publishes $d_{i,j}, i \in \{1, ..., m\}$ and $j \in \{1, ..., n\}$.

Accordingly, the participant P_j, who has got s_j, is able to compute $Q_i(x_j)$ by adding the values $f^{i-1}(s_j)$ and $d_{i,j}$ for every $i \in \{1, ...m\}$. Then, to construct a secret S_i in a stage, a set of t participants interpolate the polynomial $Q_i(x)$ by the help of values $Q_i(x_j)$ for t arbitrary $j \in \{1, ..., n\}$. The secret S_i is simply equal to $Q_i(0)$. In order to protect the security of unrevealed pseudo-shares and subsequently the secrets, the participants should follow a predetermined order for the secret recovery: the secret S_{m-l+1} should be recovered in the l-th stage, $l \in \{1, ..., m\}$ and to reconstruct the secret S_i, the participant P_j presents $f^{i-1}(s_j)$ as his/her pseudo-share.

In [14], it has been shown that the public shift values do not leak any information about the secrets and if the predetermined order of secret recovery is preserved, this scheme has computational security. However, in this scheme, the participants are able to contravene the predetermined order and compromise the security of unrecovered secrets.

The number of public values in MSSS and OSMSS schemes is an important parameter that affects the efficiency of a scheme by increasing the required memory and communication load [33]. In the above scheme, the number of public values is equal to $m \times n$. A similar MSSS scheme aiming to decrease the number of public values is presented in [16]. This scheme needs $m \times (n-t)$ public values which leads to a significant decrease of the number of public values when t is close to n. However, it still requires that the participants follow a predetermined order of secret recovery to preserve the security of the scheme.

In 1995, He and Dawson presented another MSSS scheme [15] to remove the predetermined constraint on the order of secret recovery. In this scheme, a two-variable one-way function is used instead of successively applying a one-way function on the master shares. To share the secrets in this scheme, the dealer performs the following steps.

1. Chooses n random integers $s_1, ..., s_n$ and n arbitrary constants $c_1, ..., c_n$.
2. For $i \in \{1, ..., n\}$, shares the secret S_i by applying the public shift values technique and by assuming the values $F(s_j, c_i), j \in \{1, ..., n\}$ as the pseudo-shares, where $F(.,.)$ denotes a two-variable one-way function. Let $d_{i,j} = Q_i(x_j) - F(s_j, c_i)$ be the resulting shift values.
3. Sends the master shares $s_1, s_2, ..., s_n$ through secure channels to the participants $P_1, P_2, .., P_n$, respectively and publishes $d_{i,j}, i \in \{1, ..., m\}$ and $j \in \{1, ..., n\}$.

In this scheme, different secrets could be reconstructed in different stages in an arbitrary order. Again, the reconstructed secrets and public values do not leak any information about unrecovered secrets. Furthermore, the number of public values in this scheme is $m \times n + m = m \times (n + 1)$. Note that in all of the above schemes, the dealer generates m polynomials to share m secrets.

Another MSSS scheme with similar structure is presented in [18]. In this scheme, the secrets are again assumed to be recovered in a specific order but unlike [14], this scheme could guarantee that the participants preserve this order. Similar to the previous MSSS schemes, this scheme employs m polynomials to share m secrets and it needs $m \times n$ public values. Moreover, it is a multi-use scheme, that is, even after recovering all the secrets, the master-shares of the participants would be still kept secret and hence could be used in another MSSS session.

The first OSMSS scheme presented in 2000 is based on linear block coding [22]. In this scheme, m secrets are interpreted as the m components of a message vector. This vector is then encoded using the generator matrix of a systematic block code. Finally, a number of code symbols are given to the participants as their shares and some other symbols are published. The authorized subset of participants could decode the codeword and recover the secrets, simultaneously. This scheme requires $(m + n - t + 1)$ public values.

Two more OSMSS schemes based on the Shamir's secret sharing scheme are proposed in [23] and [24]. In these schemes, the shares are generated by a polynomial that has all the secrets as its coefficients. The number of public values in the former scheme is equal to $(m+n-t+1)$ and $n+1$ when $m > t$ and $m \leq t$, respectively. For the latter scheme, this number is equal to $(m+n-t+1)$.

Many other OSMSS schemes have been presented that employ different methods for sharing the secrets. For example, the scheme in [25] is a verifiable OSMSS scheme with $2(n+1)$ and $2(n+1)+m-t$ public values for the case of $m \leq t$ and $m > t$, respectively and the one in [31] is a verifiable scheme based on linear recursive equations with $2(n+1)+m-t$ public values. In [29], another verifiable OSMSS scheme with $2(n+1)$ public values is presented. This scheme makes use of bilinear maps to provide the share verifiability for the participants.

Before we finish this section, we briefly explain the specific OSMSS scheme introduced in [23] as an example through which we reveal the security flaw of the OSMSS schemes in general.

Share Distribution Process. In this scheme, to share the secrets among the participants, the dealer first chooses a large enough prime number p. Then, he chooses n random values $s_1, s_2, ..., s_n \in \mathbb{Z}_p$ and sends them through secure channels to the n participants (p is chosen such that $S_1, S_2, ..., S_m \in \{1, 2, ..., p-1\}$). He also chooses a random value $r \in \mathbb{Z}_p$ and calculates the pseudo-shares $F(s_j, r), j \in \{1, ..., n\}$, where F resembles a two-variable one-way function. In the case of $m \leq t$, the dealer performs the following steps.

1. Chooses the random values $a_1, ..., a_{t-m} \in \{1, ..., p-1\}$ and generates a polynomial $Q(x) (\bmod\ p)$ of degree $t-1$ according to

$$Q(x) = S_1 + S_2x + ... + S_mx^{m-1} + a_1x^m + a_2x^{m+1} + ... + a_{t-m}x^{t-1} (\bmod\ p) \quad (5)$$

2. Calculates the values $y_j = Q\big(F(s_j, r)\big), j \in \{1...n\}$.
3. Publishes $(r, y_1, ..., y_n)$. The number of public values in this case is equal to $(n+1)$.

In the case of $m > t$, the dealer does the following steps.

1. Generates a polynomial $Q(x) (\bmod\ p)$ of degree $m-1$ according to

$$Q(x) = S_1 + S_2x + ... + S_mx^{m-1} (\bmod\ p) \quad (6)$$

2. Calculates the values $y_j = Q\big(F(s_j, r)\big), j \in \{1, ..., n\}$.
3. Computes the public values $Q(i) (\bmod\ p), i \in \{1, ..., m-t\}$ and publishes the vector $(r, y_1, ..., y_n, Q(1), ..., Q(m-t))$. The number of public values in this case is equal to $(n+m-t+1)$.

Secret Reconstruction Process. In the case of $m \leq t$, at least t participants could obtain t points of the form $\big(F(s_j, r), y_j\big)$ on the polynomial $Q(x)$ and using the Lagrange interpolation, they could calculate $Q(x)$ and subsequently the secrets. In the other case, in order to reconstruct the polynomial $Q(x)$ of degree $m-1$, these participants could use the $m-t$ public points $(i, Q(i)), i \in \{1, ..., m-t\}$ on the polynomial, in addition to their shares.

Although in the above scheme, $t-1$ participants could not reconstruct the m secrets, according to the following discussion, they could reduce the search space of the secrets from $|\mathcal{S}|^m$ to $|\mathcal{S}|$, where $\mathcal{S}$ denotes the secret space. Assume $m \leq t$ and consider the following set of $(t-m+1)$ equations with t unknowns $(S_1, S_2, ..., S_m, a_1, ..., a_{t-m})$.

$$y_j = S_1 + S_2F(s_j, r) + ... + S_m\big(F(s_j, r)\big)^{m-1} + a_1\big(F(s_j, r)\big)^m + ... + a_{t-m}\big(F(s_j, r)\big)^{t-1}, \quad j \in \{1, ..., t-m+1\} \quad (7)$$

Every set of $(t-m+1)$ participants, with the help of their shares, are able to omit $t-m$ unknown coefficients $a_1, ..., a_{t-m}$ and reduce the equations in (7) to one equation in m unknowns $(S_1, S_2, ..., S_m)$. Using a similar approach, $t-1$ participants could find $(t-1)-(t-m) = m-1$ linear relations between the m secrets. As a result, the search space of the secrets reduces to that of one secret for these unauthorized set of $t-1$ participants. That is, if they could find the value of just one secret somehow, they would recover all the secrets. The same argument satisfies for the case of $m > t$. In the next section, we show that what is stated here as a security flow for the OSMSS scheme in [23] is generally true for all OSMSS schemes.

4 Security flaw in OSMSS schemes

In this section, we show that in a (t, n)-threshold OSMSS scheme, an unauthorized subset of k participants ($k < \min(m, t)$) could reduce the search space of secrets from $|\mathcal{S}|^m$ to $|\mathcal{S}|^{t-k}$, where $\mathcal{S}$ and m are the secret space and the number of secrets, respectively. This is mainly because the secrets are supposed to be revealed, simultaneously. As a result, each participant uses the same share value to recover all the secrets.

Assume that at least t participants are needed to recover all the secrets and $k \in \{1, ..., \min(m, t)\}$ is an integer. The secrets and the shares are chosen randomly, according to an identical distribution from the same space $\mathcal{S} = \mathcal{S}_p$, where $\mathcal{S}_p$ is the share space. Hence,

$$H(S_1) = H(S_2) = ... = H(S_m) = H(s_1) = ... = H(s_n) \quad (8)$$

where H(.) is the Entropy function. Then,

$$\begin{aligned} I\ &(s_1, ..., s_k; S_1, ..., S_m | s_{k+1}, ..., s_t) = \\ H\ &(s_1, ..., s_k | s_{k+1}, ..., s_t) - H(s_1, ..., s_k | s_{k+1}, ..., s_t, S_1, ..., S_m) = \\ H\ &(S_1, ..., S_m | s_{k+1}, ..., s_t) - H(S_1, ..., S_m | s_1, ..., s_k, s_{k+1}, ..., s_t) \end{aligned} \quad (9)$$

With due attention to the definition of OSMSS schemes, $H(S_1, ..., S_m | s_1, ..., s_k, s_{k+1}, ..., s_t) = 0$. Therefore,

$$\begin{aligned} H\ &(S_1, ..., S_m | s_{k+1}, ..., s_t) \leq \\ H\ &(s_1, ..., s_k | s_{k+1}, ..., s_t) \leq \\ H\ &(s_1, ..., s_k) \leq H(s_1) + ... + H(s_k) = \\ kH\ &(s_1) = kH(S_1) \end{aligned} \quad (10)$$

Equation (10) shows that the uncertainty about the m secrets in $t-k$ shares is at most equal to the uncertainty of k secrets. In the case of $k = 1$, the uncertainty of $t-1$ participants about the secrets is equal to that of one secret.

5 Efficient MSSS schemes

In the previous sections, we fully investigated MSSS and OSMSS schemes. The study of OSMSS schemes revealed that they have an intrinsic security weakness that causes an authorized set of participants to be able to reduce the uncertainty about the secrets to that of one secret. That is, in a (t, n)-threshold OSMSS scheme, every set of $t-1$ participants would be able to find some relations between their shares and the secrets and subsequently recover all the secrets, if they could somehow seize the value of just one secret. Moreover, OSMSS schemes provide less flexibility, compared to MSSS. More precisely, in OSMSS schemes, all the secrets get revealed simultaneously which causes the required number of public values in these schemes to be much lower than the corresponding number in MSSS schemes. The required number of public values in all of OSMSS schemes studied so far, depends linearly on the number of participants (n) and the number of secrets (m) which is equal to $m + n - t + 1$ in the best case [22,24].

MSSS schemes are more flexible, that is, in these schemes, the secrets could be recovered independently and possibly according to a desired order in successive stages. The required number of public values in all of the proposed MSSS schemes depends on the product of the number of participants and the number of secrets whose minimum value has been equal to $m \times (n - t)$ [16].

In this section, we propose two threshold MSSS schemes based on a linear one-way function. The first scheme needs just m public values in order to share m secrets among n participants. Then, by employing bilinear maps, we propose a verifiable version of this scheme that needs $2(n + 1) + m - t$ public values to share m secrets according to a (t, n)-threshold MSSS scheme. As referred to in section 3, no verifiable MSSS scheme has been proposed so far and the verifiable OSMSS schemes in [31] and [29] require $2(n + 1) + m - t$ and $2(n + 1)$ public values, respectively.

5.1 Preliminaries

Bilinear maps and linear one-way functions were first introduced in [34] and [30], respectively and then found some applications in identity-based cryptosystems [34] and network security [35], [36]. In this section, we briefly introduce these concepts, before we employ them in the proposed MSSS schemes.

Definition 1 [34] Let $\mathbb{G}$ be an additive group and $\mathbb{G}_1$ be a multiplicative group, both of order q for some large prime q (for example 160 bits). A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is said to be an admissible bilinear map, if

1. $e(aP, bQ) = e(P, Q)^{ab}$ for all $a, b \in \mathbb{Z}_q$ and $P, Q \in \mathbb{G}$ (Bilinear condition).
2. The map does not project all elements of $\mathbb{G} \times \mathbb{G}$ to the identity element of $\mathbb{G}_1$ (Non-degeneracy condition).
3. There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$ (Computability condition).

An example of the groups $\mathbb{G}$ and $\mathbb{G}_1$ together with a bilinear map could be found in [34].

The existence of an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ leads to the following result in the groups $\mathbb{G}$ and $\mathbb{G}_1$ [34].

- The Decision Diffie-Hellman problem (DDH) in $\mathbb{G}$ is easy, that is, it is easy to distinguish between the values (P, aP, bP, abP) and (P, aP, bP, cP) where a, b, c are random in $\mathbb{Z}_q^*$ and P is random in $\mathbb{G}^*$ but the Computational Diffie-Hellman problem (CDH) in $\mathbb{G}$ can be still hard; in other words, it is hard to find abP given random values (P, aP, bP).

Also, the discrete logarithm problem on elliptic curves, as defined below, is supposed to be hard [34].

- Let P be a generator and Q be an arbitrary element of $\mathbb{G}$. There exist an integer $r \in \mathbb{Z}_q^*$ such that $Q = rP$. The discrete logarithm problem on elliptic curves is equivalent to finding r, given P and Q.

Remark 1 Consider the isomorphism induced from $\mathbb{G}$ to $\mathbb{G}_1$ by the bilinear map e. More specifically, for a point $Q \in \mathbb{G}^*$ define the isomorphism $f_Q : \mathbb{G} \to \mathbb{G}_1$ by $f_Q(P) = e(P, Q)$. The existence of an efficient algorithm for inverting f_Q, for some Q, would lead to an efficient algorithm for solving discrete logarithm problem in $\mathbb{G}_1$ [34]. Consequently, the isomorphism f_Q is believed to be a one-way function whenever discrete logarithm is believed to be hard in $\mathbb{G}_1$, as it is the case in all of the examples given in [34]. Therefore, throughout the paper the bilinear map e is considered as a one-way function (P and Q cannot be inferred from $e(P, Q)$).

In [30], a linear one-way function $h : \mathbb{G} \times \mathbb{Z}_q^* \to \mathbb{G}$ is introduced to be used in an identity-based encryption (IBE) system. Here, we benefit from the properties of h in the proposed scheme.

Definition 2 [30] Let $\mathbb{G}$ be an additive group of order q, where q is a large prime number. $h : \mathbb{G} \times \mathbb{Z}_q^* \to \mathbb{G}$ is a linear one-way function, if

- For all $P \in \mathbb{G}$ and $a, x \in \mathbb{Z}_q^*$ we have $h(aP, x) = ah(P, x)$.
- Given $x, x_i \in \mathbb{Z}_q^*, P \in \mathbb{G}$ and $\big(x_i, h(aP, x_i)\big)$ for all $i \in \{1, ..., n\}$, $h(aP, x)$ could not be computed, using any probabilistic polynomial-time algorithm.

The function h, defined above, is a one-way function with respect to its first argument, that is, P cannot be inferred from $h(P, x)$ and x.

5.2 An MSSS scheme based on a linear one-way function

Let $h : \mathbb{G} \times \mathbb{Z}_q^* \to \mathbb{G}$ be a linear one-way function where q and $\mathbb{G}$ denoted a sufficiently large prime number and an additive group of order q, respectively. Here, we assume that $\mathbb{G}$ is an additive group that consists of the points on an elliptic curve. Also, assume that the secrets $S_1, S_2, ..., S_m$ are all in $\mathbb{G}$. The proposed MSSS scheme consists of two processes: share distribution and secret reconstruction.

Share Distribution Process. To share the secrets $S_1, S_2, \cdots, S_m$ according to a (t, n)-threshold MSSS scheme, the dealer executes the following steps.

1. Chooses random coefficients $a_0, a_1, ..., a_{t-1} \in \mathbb{Z}_q^*$ and generates the polynomial $Q(x) = a_0 + a_1 x + ... + a_{t-1}x^{t-1}$.
2. Computes the values $s_1 = Q(x_1), s_2 = Q(x_2), \cdots, s_n = Q(x_n)$, where $x_1, ..., x_n \in \mathbb{Z}_q^*$ are the label values assigned to the participants $P_1, ..., P_n$, respectively.
3. Randomly chooses a generator $P \in \mathbb{G}$ and sends $s_1P, s_2P, ..., s_nP$ via secure channels to the participants $P_1, P_2, ..., P_n$ as their shares, respectively.
4. Calculates the values $\alpha_1 = h(s_0P, 1), \alpha_2 = h(s_0P, 2), \cdots, \alpha_m = h(s_0P, m)$, where $s_0 = Q(0)$. Next, he publishes the values $S_1 - \alpha_1, ..., S_m - \alpha_m$.

Secret Reconstruction Process. Assume that a set of t participants $P_1, ..., P_t$ want to collaborate in one stage to recover the secret $S_i, i \in \{1, ..., m\}$. Each of these participants first calculates and pools his/her pseudo-share $h(s_jP, i), j \in \{1, ..., t\}$. Next, they compute the value $h(s_0P, i)$ from the following equation.

$$\begin{aligned} &\textstyle\sum_{j=1}^{t} \Bigg(\prod_{\substack{i=1\\ i\neq j}}^{t} \frac{x_i}{x_i - x_j}\Bigg) h(s_jP, i) \\ &= h\Bigg(\big(\sum_{j=1}^{t} s_j \prod_{i=1, i\neq j}^{t} \frac{x_i}{x_i - x_j}\big)P, i\Bigg) \\ &= h\big(Q(0)P, i\big) = h(s_0P, i) \qquad (11) \end{aligned}$$

The participants $P_1, ..., P_t$ can then recover the secret S_i from $S_i - \alpha_i$ and $\alpha_i = h(s_0P, i)$.

The number of public values for sharing m secrets among n participants in the proposed scheme is independent of n and it is equal to m. Moreover, the participants can reconstruct the secrets according to their desired order rather than in a prespecified one.

5.3 A verifiable MSSS scheme based on a linear one-way function and a bilinear map

In this section, we present a modified version of the proposed MSSS scheme. In the new scheme, the dealer does not need secure channels to assign the shares to the participants. Moreover, the participants who collaborate in a secret reconstruction stage could verify the correctness of pseudo-shares that are presented by the other participants, before recovering the secrets. Here, in addition to the linear one-way function h, we employ a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ to verify the shares, where $\mathbb{G}$ and $\mathbb{G}_1$ are defined as in section 5.1.

Share Distribution Process. The dealer randomly chooses a generator $P \in \mathbb{G}$ and a random secret value $r \in \mathbb{Z}_q^*$. Then, he computes $Q = rP$ and publishes the values P and Q. Subsequently, each participant randomly chooses a secret value $s_j \in \mathbb{Z}_q^*$ as his/her share and sends the value s_jQ to the dealer, via public channels. Upon receiving all $s_jQ, j \in \{1, ..., n\}$, the dealer first checks if these values are all distinct and then he computes the values $s_jP, j \in \{1, ..., n\}$ by the following equation.

$$r^{-1}(s_jQ) = r^{-1}(s_jrP) = s_j(r^{-1}rP) = s_jP \qquad (12)$$

Then, the dealer computes the value $Q(0)P$ according to the following equation, where $Q(x)$ is a polynomial of degree at most $n-1$ which passes through the points $(x_1, s_1P), (x_2, s_2P), ..., (x_n, s_nP)$.

$$Q(0)P = \sum_{j=1}^{n} \Bigg(\prod_{\substack{i=1\\ i\neq j}}^{n} \frac{x_i}{x_i - x_j}\Bigg) s_jP \qquad (13)$$

The dealer also computes $Q(d_k)P$ as shown in (14), where $d_k, k \in \{1, ..., n-t\}$ are the $n-t$ smallest values in the set $\{1, ..., q-1\} \setminus \{x_j | j = 1, ..., n\}$ and then calculates $\alpha_1 = h\big(Q(0)P, 1\big), \alpha_2 = h\big(Q(0)P, 2\big), ..., \alpha_m = h\big(Q(0)P, m\big)$. Finally, he publishes the vector $\big(s_iQ, S_j - \alpha_j, Q(d_k)P\big)_{i=1,...,n, j=1,...,m, k=1,...,n-t}$

$$\begin{aligned} Q(d_k)P = \sum_{j=1}^{n} \Bigg(\prod_{\substack{i=1\\ i\neq j}}^{n} \frac{d_k - x_i}{x_j - x_i}\Bigg) s_jP, \qquad (14) \\ k \in \{1, ..., n-t\}. \end{aligned}$$

Secret Reconstruction Process. Let $P_1, P_2, ..., P_t$ resemble t participants who come together in one stage to recover the secret $S_i, i \in \{1, ..., m\}$. Using their pseudo-shares $h(s_jP, i)$, $j \in \{1, ..., t\}$ and the public values $Q(d_1)P, Q(d_2)P, ..., Q(d_{n-t})P$, the participants are able to calculate α_i according to

$$\begin{aligned} \alpha_i &= h(Q(0)P, i) \\ &= \sum_{j=1}^{t} \Big(\prod_{i=1, i\neq j}^{t} \frac{x_i}{x_i - x_j}\Big)\Big(\prod_{i=1}^{n-t} \frac{d_i}{d_i - x_j}\Big) h(s_jP, i) \\ &+ \sum_{j=1}^{n-t} \Big(\prod_{i=1, i\neq j}^{n-t} \frac{d_i}{d_i - d_j}\Big)\Big(\prod_{i=1}^{t} \frac{x_i}{x_i - d_j}\Big) h(Q(d_j)P, i) \qquad (15) \end{aligned}$$

Now, the secret S_i is simply the sum of α_i and $S_i - \alpha_i$.

Share Verification Process. Before computing α_i and recovering the secret S_i, the participants $P_1, P_2, ..., P_t$ could confirm the correctness of the submitted pseudo-shares through the following equation,

$$e(h(s_jP, i), Q) = e(h(P, i), s_jQ),\ 1 \le i \le m,\ j = 1, ..., t, \quad (16)$$

where $s_jQ, j \in \{1, ..., n\}$ are public values.
If the equality in (16) holds for any $j \in \{1, ..., t\}$, the participants get confident about the validity of the pseudo-share $h(s_jP, i)$.

The proposed verifiable MSSS scheme has $2(n+1)+m-t$ public values, including $Q, P, s_1Q, ..., s_nQ, S_1 - \alpha_1, ..., S_m - \alpha_m, Q(d_1)P, ..., Q(d_{n-t})P$. Similar to the scheme presented in section 5.2, every t participants in this scheme are able to recover the secrets in a desired order and in consecutive stages.

A comprehensive analysis of the above schemes is presented in the next section.

6 Analysis of the proposed MSSS schemes

In this section, we try to present a thorough analysis of the MSSS schemes, presented in section 5. This analysis is divided into two parts. First, we investigate the security of the proposed schemes. Next, we compare these schemes with the other MSSS schemes already presented in the literature.

We prove the computational security of the proposed MSSS schemes, by means of a couple of theorems. First, consider the scheme presented in Section 5.2. In that scheme, the random generator $P \in \mathbb{G}$ does not need to be published. The following theorems, however, are still valid in the case that P is public.

Theorem 1 *In the MSSS scheme based on a linear one-way function, one cannot gain any information about the undisclosed secrets from already recovered secrets, in polynomial time.*

Proof: Assume that the secrets $\{S_i\}_{i\in I}, I \subset \{1, ..., m\}$ are recovered. Consequently, the values $\{\alpha_i = h(s_0P, i)\}_{i\in I}$ and the pseudo-shares $\{h(s_jP, i), 1 \le j \le n\}_{i\in I}$ are revealed to those t participants. Regarding the properties of the one-way linear functions,stated in Definition 2 of Section 5.1, however, none of the values $s_0P, h(s_0P, k),\ s_jP$ and $h(s_jP, k), 1 \le j \le n$ for $k \notin I$ could be calculated from $\{\alpha_i = h(s_0P, i)\}_{i\in I}$, $\{h(s_jP, i), 1 \le j \le n\}_{i\in I}$ and I, in polynomial time. Hence, no information about undisclosed secrets is leaked from the recovered secrets, in polynomial time. ■

Theorem 2 *In the MSSS scheme based on a linear one-way function, no subset of $P = \{P_1, P_2, ..., P_n\}$ with at most $t-1$ participants obtain any information about the secrets $S_i, i \in \{1, ..., m\}$, in polynomial time.*

Proof: Assume that a group of $t-1$ participants conspire to recover the secret S_i in a stage. According to Theorem 1, the recovered secrets do not leak any information about the undisclosed secret S_i in polynomial time. Without loss of generality we assume that S_i is the first secret to be recovered. To calculate S_i, the cheating participants need one of the values s_0P or $h(s_0P, i)$. Calculation of s_0P requires knowledge of at least t values of the master-shares $s_jP, 1 \le j \le n$ on the polynomial $Q(x)P$, while these participants have only $(t-1)$ out of the n master-shares. Regarding the perfectness of Shamir's scheme, having only $t-1$ values of s_iP on the polynomial $Q(x)P$ with the coefficients chosen randomly according to a uniform distribution from $\mathbb{Z}_q$, it is not possible to obtain any information about s_0P, the constant term of this polynomial. Moreover, to compute $h(s_0P, i)$, these participants need at least t values of the form $h(s_jP, i), 1 \le j \le n$ but they just have $t-1$ of them. Hence, the cheating participants do not extract any information about the secret S_i. Bear in mind that the public values in this scheme are of the type of shift values presented in [14] and as shown in the same paper, these values do not leak any information about the secrets. ■

The above theorems confirm that the proposed MSSS scheme based on a linear one-way function has the computational security. Moreover, it is a multi-use scheme. This means that even after recovering all of the secrets, the master-shares of the participants (that is, $s_jP, 1 \le j \le n$) are kept confidential and the shareholders can use their shares to recover a new set of secrets.

Security analysis of the verifiable MSSS scheme, introduced in Section 5.3, is similar to the above analysis. The only point that should be considered is that the values published to provide verifiability of the shares do not give any information about the shares of the participants. This statement is proved in the next theorem.

Theorem 3 *In the verifiable MSSS scheme based on a linear one-way function $h : \mathbb{G} \times \mathbb{Z}_q^* \to \mathbb{G}$ and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, no information about the master-shares leaks from the public values in polynomial time.*

Proof: It suffices to show that having the values $P, Q, s_jQ, 1 \le j \le n$, one could not compute the values s_j and s_jP. With respect to the discrete logarithm problem, it is computationally impossible to calculate s_j from s_jQ and Q. Similarly, the participants could not compute $r, r^{-1} \in \mathbb{Z}_q^*$ which satisfy $Q = rP$ and $P = r^{-1}Q$. As a result, none of the participants can compute s_jP from $s_jP = r^{-1}s_jQ$. Consequently, it is not possible for the participants to calculate any of s_j or s_jP from the public values. This completes the proof. ■

Table 6 shows the results of comparing the two MSSS schemes, proposed in section 5, with previous schemes from the following points of view: 1) The level of security; 2) Verifiability of the shares; 3) Number of public values for sharing m secrets among n participants according to a (t, n)-threshold scheme. It can be inferred from the results that the proposed MSSS scheme based on a linear one-way function needs the least number of public values; moreover, the verifiable MSSS scheme based on a bilinear map requires nearly the same number of public values as the OSMSS schemes in [28],[29]. This is while the two latter schemes fail to provide the computational security.

7 Conclusions

In this paper, we have considered two generalizations of a secret sharing scheme: The One-Stage Multisecret Sharing scheme and the Multi-Stage Secret Sharing scheme. The desired level of security in these schemes is the computational security. However, we have proved that an OSMSS scheme fails to provide this security level. More precisely, in a (t, n) threshold OSMSS scheme with m secrets, a set of $k < t$ unauthorized participants could reduce the uncertainty about the m secrets to that of only $t-k$ secrets. This is mainly because each participant applies the same share value in recovering

Table 1 Comparison of the two proposed MSSS schemes with the previous schemes

Scheme	Computational Security	Verifiability	No. of Public Values
OSMSS [22]	No	No	$n+m-t+1$
OSMSS [23]	No	No	$n+1(m \leq t)$ $n+m-t+1(m>t)$
OSMSS [24]	No	No	$m+n-t+1$
OSMSS [27]	No	No	$n+1(m \leq t)$ $n+m+1(m>t)$
OSMSS [28]	No	Yes	$2(n+1)+m-t$
OSMSS [29]	No	Yes	$2n+1$
MSSS [14]	Yes	No	$m \times n$
MSSS [15]	Yes	No	$m \times (n+1)$
MSSS [16]	Yes	No	$m \times (n-t)$
MSSS [18]	Yes	No	$m \times n$
MSSS [20]	Yes	No	$m \times (n-t+1)$
The proposed MSSS scheme	Yes	No	m
The proposed verifiable MSSS scheme	Yes	Yes	$2(n+1)+m-t$

different secrets. Alternatively, in an MSSS scheme, the secrets could be recovered in different stages. To make it happen, the participants should derive some pseudo-shares from their master-shares to present in recovery of different secrets.

OSMSS and MSSS schemes come usually with a number of public values that are used for the share verification and the secret reconstruction. The study of MSSS schemes revealed that the number of public values in the proposed schemes is proportional to product of the number of the participants and that of the secrets and in the best case, it is equal to $m \times n$. By employing a linear one-way function, we have presented an MSSS scheme where the number of public values exclusively depends on the number of secrets. This scheme uses only one generating polynomial for sharing different secrets and hence has less computational complexity in the share distribution process, when compared to the previously proposed schemes. The new scheme is multi-use and provides the desired computational security. Finally, using bilinear maps, we have presented a verifiable version of our scheme in which the participants are able to verify the released shares from other participants. This verifiable scheme does not need any secure channel and the participants generate their own shares. The number of public values in this scheme is equal to $2(n+1)+m-t$.

Acknowledgements This work is partially supported by ITRC under the contraction no. T500/20961 and supported by cryptography chair of Iran NSF.

References

1. A. Shamir, "How to Share a Secret", Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.
2. G.R. Blakley, "Safeguarding Cryptographic Keys", AFIPS, National Computer Conference, vol. 48, pp. 313–317 , 1979.
3. M. Mignotte, "How to Share a Secret", Cryptography Proceedings, Burg Feuerstein 1982, T. Beth, ed., LNCS 149, pp. 371–375, 1983.
4. J. Benaloh and J. Leichter, "Generalized Secret Sharing and Monotone Functions", Advances in Cryptology – CRYPTO '88, S. Goldwasser, ed., LNCS 403, pp. 27–35, 1989.
5. B. Chor and S. Goldwasser and S. Micali and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults", Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science, IEEE Press, pp. 383–395, 1985.
6. M. Stadler, "Publicley Verifiable Secret Sharing", Advances in Cryptology, EUROCRYPT'96, LNCS, vol. 1070, pp. 190–199, 1996.
7. E.F. Brickell and D.R. Stinson, "The Detection of Cheaters in Threshold Schemes", Society for Industrial and Applied Mathematics (SIAM), vol. 4, no.4, pp. 502–520, 1991.
8. W. Ogata and K. Kurosawa, "Optimum Secret Sharing Scheme Secure against Cheating", U. Maurer (ed.) Advances in Cryptology, EUROCRYPT'96. LNCS, vol. 1070, pp. 200–211. Springer, Heidelberg, 1996.
9. K.M. Martin and R. Safavi-Naini and H. Wang, "Bound and Techniques for Efficient Redistribution of Secret Shares to New Access Structures", Computer Journal, vol. 42, no. 8, pp. 638–649, 1999.
10. S.G. Barwick and W.A. Jackson and K.M. Martin, "Updating the Parameters of a Threshold Scheme by Minimal Broadcast", IEEE Transactions On Information Theory, vol. 51, no. 2, pp. 620–633, 2005.
11. E.D. Karnin and J.W. Greene and M.E. Hellman, "On Secret Sharing System", IEEE Transaction on Information Theory, vol. 29, no. 1, pp. 35–41, 1983.
12. E.F. Brickell and D.M. Davenport, "On the Classification of Ideal Secret Sharing Schemes", Journal of Cryptology, vol. 4, pp. 123–134, 1991 [Preliminary version appeared in Advances in Cryptology – CRYPTO '89, G. Brassard, ed., LNCS 435, pp. 278–285, 1990].
13. H. Krawczyk, "Secret Sharing Made Short", Advances in Cryptology – CRYPTO '93, D. R. Stinson, ed., LNCS 773, pp. 136–146, 1994.

14. J. He and E. Dawson, "Multi-Stage Secret Sharing Scheme Based on One-way Function", Electronic Letters, vol. 30, no. 19, pp. 1591–1592, 1994.
15. J. He and E. Dawson, "Multisecret-Sharing Scheme Based on One-way Function", Electronic Letters, vol. 31, no. 2, pp. 93–95, 1995.
16. L. Harn, "Comment: Multistage Secret Sharing based on One-way Function", Electronics Letters, vol. 31, no. 4, pp. 262–262, 1995.
17. L. Harn, "Efficient Sharing (Broadcasting) of Multiple Secrets", Proceeding of the IEE Comput. Digit. Tech., vol. 142, no. 3, pp. 237–240, May 1995.
18. T.Y. Chang and M.S. Hwang and W.P. Yang, "A New Multi-Stage Secret Sharing Scheme Using One-Way Function", ACM SIGOPS Operating Systems, vol. 39, pp. 48–55, 2005.
19. C.W. Chan and C.C. Chang, "A Scheme for Threshold Multi-Secret Sharing", Applied Mathematics and Computation, vol. 166, pp. 1–14, 2005.
20. H.X. Li and C.T. Cheng and L.J. Pang, "An Improved Multi-Stage (t,n)-Threshold Secret Sharing Scheme", WAIM05, Fan W., Wu Z., and Yang J., eds., LNCS 3739, pp. 267–274, 2005.
21. M. Fatemi and T. Eghlidos and M. Aref, "A Multi-stage Secret Sharing Scheme Using All-or-Nothing Transform Approach", ICICS'09, LNCS 5927, pp. 449–458, 2009.
22. H.Y. Chien and J.K. Jan and Y.M. Tseng, "A Practical (t, n) Multi-Secret Sharing Scheme", IEICE Transactions on Fundamentals, vol. E83-A, no. 12, pp. 2762–2765, 2000.
23. C.C. Yang and C.C. Chang and M.S. Hwang, "A (t, n) multi-secret sharing scheme", Applied Mathematics and Computation, vol. 151, no. 2, pp. 483–490, 2004.
24. L.J. Pang and Y.M. Wang, "A New (t,n) Multi-Secret sharing Scheme Based on Shamir's Secret Sharing", Applied Mathematics and Computation, vol. 167, pp. 840–848, 2005.
25. J. Zhao and R. Zhao, "A Practical Verifiable Multi-Secret Sharing Scheme", Computer Standards and Interfaces, vol. 29, no. 1, pp. 138–141, 2007.
26. M.H. Dehkordi and S. Mashhadi, "An Efficient Threshold Verifiable Multi-Secret Sharing", Computer Standards and Interfaces, vol. 30, pp. 187–190, 2008.
27. S. Runhua and H. Liusheng and L. Yonglong and Z. Hong, "A Threshold Multi-Secret Sharing Scheme", IEEE International Conference on Networking, Sensing and Control, ICNSC'08, pp. 1705–1707, 2008.
28. M.H. Dehkordi and S. Mashhadi, "New Efficient and Practical Verifiable Multi-Secret Sharing Schemes", Information Sciences, vol. 178, pp. 2262–2274, 2008.
29. S.J. Wang and Y.R. Tsai and J.J. Shen, "Dynamic Threshold Multi-secret Sharing Scheme Using Elliptic Curve and Bilinear Maps", International Conference on Future Generation Communication and Networking (FGCN'08), vol. 2, pp. 405–410, 2008.
30. J. Horwitz and B. Lynn, "Toward Hierarchical Identity-Based Encryption," Proceedings of EUROCRYPT '02, LNCS 2332, pp.466–481, 2002.
31. M.H. Dehkordi and S. Mashhadi, "Verifiable Secret sharing Schemes Based on Non-homogeneous Linear Recursions and Eliptic Curves", Computer Communications, vol. 31, pp. 1777–1784, 2008.
32. C. Wei and L. Xiang and B. Yuebin and G. Xiaopeng, "A New Dynamic Threshold Secret Sharing Scheme from Bilinear Maps", ICPPW, Intl Conf. on Parallel Processing Workshops07, IEEE Computer Society, pp. 19–22, 2007.
33. G.D.L. Crescenzo, "Sharing One Secret vs. Sharing Many Secrets: Tight Bounds on the Average Improvement Ratio", Proc. of 11th Annual ACM-SIAM Symp. on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp. 273–274, 2000.
34. D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", Advances in Cryptology, Crypto'01, LNCS 2139, pp. 213–229, 2001.
35. Z. Wan and K. Ren and B. Preneel, "A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks", First ACM Conf. on Wireless Network Security (WiSec08), 2008.
36. M. Fatemi and S. Salimi and A. Salahi, "Anonymous Roaming in Universal Mobile Telecommunication System Mobile Networks", IET Information Security, vol. 4, no. 2, pp. 93-103, 2010.