Journal of Computing and Security

January 2017, Volume 4, Number 1 (pp. 3–12)
http://www.jcomsec.org

Verifiable Social Multi-Secret Sharing Secure in Active Adversarial Model

Nasrollah Pakniat [a,*] Ziba Eslami [b]

[a] *Information Science Research Department, Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Iran.*
[b] *Department of Computer and Data Science, Shahid Beheshti University, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 02 Octoboer 2017
Revised: 02 January 2018
Accepted: 23 April 2018
Published Online: 25 July 2018

Keywords:
Secret Sharing, Social Secret Sharing, Symmetric Encryption, Computational Security, Trust Model.

ABSTRACT

Social secret sharing, introduced in 2010 by Nojoumian et al., allows to share a secret among a set of participants whose authorities can vary over time. However, existing social secret sharing schemes are only capable of sharing one secret during each execution. To overcome this drawback, in this paper, we employ symmetric encryption schemes to propose a social multi-secret sharing scheme. It is proved that the proposed scheme provides computational security in the active mobile adversary. Moreover, to indicate the efficiency of the proposed scheme, comparison with existing (single) social secret sharing schemes is provided.

1 Introduction

The concept of secret sharing (SS) was introduced independently in 1979 by Shamir [1] and Blakley [2]. In an SS scheme, a dealer shares a secret among a set of participants in such a way that while later the secret can be recovered from the shares corresponding to each authorized subset of participants, non authorized subsets obtain no information about the secret from their shares. The set of authorized subsets is called the access structure of the scheme. (t, n)-threshold secret sharing (denoted for short as (t, n)-TSS) is the most theoretically studied and practically applied type of secret sharing in which the access structure consists of those subsets containing at least t players out of a total of n players. It should be noted that both of the original secret sharing schemes were (t, n)-TSS. Ordinary TSS schemes are able to share only one secret during each execution. However, many SS applications, such as those associated with key-management, require the protection of more than one secret. Executing SS schemes multiple times to separately share each of the secrets is the trivial solution to this problem, but in this case each participant should store a lot of information securely. In order to circumvent this issue, multi-SS (MSS) schemes are introduced in the literature. There are many different definitions for MSS. Here, we consider the following definition introduced by Jackson et al. [3] which is used also in several constructions ([4–10]).

A *multi-secret sharing* (MSS) scheme is a method to share more than one secret among a group of participants in such a way that:

(1) any authorized subset of participants is able to recover all the secrets,
(2) any non-authorized subset of participants obtains no information about any of the secrets.

If these conditions are met, but the knowledge on some of the secrets enables the participants in a non-authorized subset to recover information on other secrets, the scheme would be called a *weakly secure*

[*] Corresponding author.

Email addresses: `pakniat@irandoc.ac.ir` (N. Pakniat), `z_eslami@sbu.ac.ir` (Z. Eslami)

https://dx.doi.org/10.22108/jcs.2018.108395
ISSN: 2322-4460 © 2017 JComSec. All rights reserved.

MSS scheme. Otherwise, we have a *strongly secure* MSS scheme.

Existing MSS schemes can be classified into two categories: multistage SS schemes and single-stage MSS schemes. In the case of multistage SS, secrets can be recovered in different stages without jeopardizing the security of the uncovered secrets [11–15]. Compared to the multistage SS schemes, single-stage MSS (which is the focus of this research and will be denoted hereafter only by MSS) schemes are more efficient. In this type of MSS, all the secrets will be recovered at once. A practical MSS scheme proposed in 2000 by Chien et al. [16] by using systematic block codes and matrices. Another practical MSS scheme is proposed in 2004 by Yang et al. [17] based on Shamir's SS scheme. In the above-mentioned schemes, the dealer and participants can provide fake shares to other players. To prevent the participants and the dealer from cheating, verifiable MSS (VMSS) schemes can be used [18]. There is also a generic construction to convert any SS scheme to an MSS scheme using cellular automata in [19]. However, the complexity of this conversion is quadratic in the number of the secrets. Therefore, this approach is not practical in all settings.

Some issues can be considered in all of the above-mentioned (M)SS schemes:

(1) The shared secret is only secure against the static adversaries, *i.e.*, those who can not get access to the shares corresponding to an authorized subset of participants in the secret's lifetime. However, some secrets with long lifetime require to be secure against mobile adversaries that over time can get access to the shares corresponding to subsets of participants of their choices.
(2) During reconstruction of the secret, all participants have the same level of authority. This is not a realistic assumption and it should be possible to assign different authorities to participants due to the difference in their reputations or other participants' trusts in them.
(3) Participants' authorities are fixed over time. Again, this is not a realistic assumption and participants' authorities can change over time due to the changes in their reputations or other participants' trusts in them.

To solve the first problem, the notion of proactive secret sharing (PSS) is introduced by Herzberg et al. in [20]. To the best of our knowledge, there exist only a few PSS scheme in the literature [20–24]. However, none of them can be used to share multiple secrets simultaneously. As the solution to the second problem, multipartite secret sharing schemes can be used in which there exists different ways to provide different authorities. The interested readers may refer to [25–31] for examples of some multipartite SS schemes. Almost all of these schemes can be easily extended to an MSS variant (by using the same ideas used in ordinary TSS). There are also some researches that tried to propose more efficient multipartite MSS schemes [4, 32, 33].

To solve the last problem, the concept of social secret sharing (SSS) is introduced in [34] by Nojoumian et al. The authors of [34] also proposed two constructions for SSS schemes with unconditional security. Their first scheme provides security in the presence of passive adversaries who try to obtain information about the secret by eavesdropping. Their second scheme withstands adversaries who can actively interfere with the scheme, as opposed to just observing and analyzing. In [35], Eslami et al. employed hierarchical threshold secret sharing and proposed another social secret sharing scheme with unconditional security in passive adversarial model.

1.1 Our Contribution

Unfortunately, none of the existing SSS schemes can be used for simultaneous sharing of multiple secrets. This is in fact due to the fact that here, the shares need to be updated proactively. To fill this gap, the aim of this paper is to apply symmetric encryption to Nojoumian et al.'s SSS scheme and obtain a social MSS (SMSS) scheme. However, the problem is not as trivial as it seems and naive application of symmetric encryption would result in an inefficient SMSS scheme. That is because Nojoumian et al.'s SSS scheme is unconditionally secure and its unconditional security is obtained at the cost of some communication and computational overheads. However, computational security is the best that can be achieved when we are dealing with threshold MSS [36, 37] or multi-use symmetric encryption concepts. Therefore, here is a list of our contribution to solve this problem:

- We propose the first construction of a social secret sharing scheme for sharing multiple secrets (*i.e.*, SMSS) by exploiting symmetric encryption methods as well as Feldman's commitment scheme [38] together with Nojoumian et al.'s idea.
- Our share renewal process improves that of Nojoumian et al.'s scheme in terms of the communication costs.
- We prove that our scheme achieves strong security in the presence of a mobile active adversary.
- We provide comparisons to show the efficiency of our SMSS scheme.

1.2 Organization of the paper

The rest of the paper is organized as follows: in Section 2, the preliminaries needed in the rest of the paper are reviewed. The notion of social multi-secret sharing and the proposed scheme are presented in Section 3. Section 4 analyzes the security and efficiency of the proposed scheme. Finally, the concluding remarks are provided in Section 5.

2 Preliminaries

In this section, the concepts of social secret sharing and symmetric encryption are reviewed.

2.1 Social Secret Sharing

A social secret sharing scheme is defined by three algorithms; "sharing" (*Sha*), "social tuning" (*Tun*) and "reconstruction" (*Rec*) algorithms. In *Sha*, the dealer shares a secret among a group of participants with different authorities and then leaves the scheme. *Tun* is periodically performed after the sharing phase. Its aim is to adjust the participants' authorities based on their behaviors (cooperation/availability) over time using a trust function [39]. Newcomers are always able to join the scheme and receive shares of the secret through this algorithm. There would be no necessity for the presence of the dealer and authorized subsets of participants can execute *Tun* without revealing the secret. When all participants in an authorized subset decide to reconstruct the secret, the *Rec* algorithm is executed to recover the secret. For further clarification and details of social secret sharing, the interested reader is referred to [34].

2.2 Symmetric Encryption

A symmetric encryption scheme is a set of three polynomial-time algorithms (*Gen*, *Enc*, *Dec*) such that *Gen* takes a security parameter α in unary and returns a secret key k; *Enc* takes a key k and a message m and returns a ciphertext c; *Dec* takes a key k and a ciphertext c and returns m if k was the key under which c was produced.

There are many definitions for the security of a symmetric encryption scheme. In this paper, the following security definition is required (Note that it may not be adequate for a symmetric encryption scheme to be considered secure, but it is sufficient for our goal).

A symmetric encryption scheme is secure against known plaintext attack (KPA) (and would be called KPA-secure) if, there exists no polynomial-time adversary able to obtain any information about the plaintext that corresponds to a randomly generated ciphertext even by accessing one or more pairs of plaintext/ciphertext encrypted under the same key.

3 The Proposed Social Multi-Secret Sharing Scheme

In this section, first, the notion of social secret sharing is extended to social multi-secret sharing (SMSS). Then, by using symmetric encryption schemes, Nojoumian et al.'s SSS scheme and Feldman's VSS scheme, we construct an efficient SMSS scheme.

3.1 Social Multi-Secret Sharing

An SMSS scheme is a social secret sharing scheme in which the simultaneous sharing of multiple secrets is possible. The same as an SSS scheme, an SMSS scheme is defined by three algorithms; "sharing" (*Sha*), "social tuning" (*Tun*) and "reconstruction" (*Rec*) algorithms. The definitions of all the algorithms are as before except that the input of *Sha* and the output of *Rec* algorithms are both a set of secrets. For the security, the definition of strongly secure MSS scheme (explained in Section 1) is considered here.

3.2 Notations

We use the following notations to describe the scheme.

U: the set of participants,
t: the threshold parameter,
n: number of participants,
P_i: the ith participant,
w_i: the weight assigned to P_i,
m: number of secrets,
S_i: the ith secret,
Π: a KPA-secure symmetric encryption scheme,
sk: the secret key,
c_i: the ith ciphertext,
sh_{ij}: the jth share of P_i,
$sh_{ij \to kl}$: the lth sub-share of P_k obtained from P_i's jth share.

3.3 The Proposed Scheme

To the best of our knowledge, none of the existing approaches used to obtain MSS schemes (except for the one of [19]) are applicable when the participants' shares required to be updated proactively. The only applicable solution is to use the idea of [19] to convert existing SSS schemes to SMSS schemes. However, the computational overhead of this approach is quadratic in the number of the secrets and therefore, it is not efficient. Our motivation here is to apply symmetric encryption to Nojoumian et al.'s SSS scheme and obtain an efficient SMSS scheme. The same as all existing threshold MSSS, our scheme achieves computational security.

Let $U = \{P_1, P_2, \cdots, P_n\}$ be the set of n participants and let $\{S_1, S_2, \cdots, S_m\} \subset Z_q^m$ denote a set of m secrets where q denotes a large prime number. Assume $\Pi = (Gen,\ Enc,\ Dec)$ is a KPA-secure symmetric encryption scheme.

In the sharing algorithm of the proposed scheme, at first, the dealer runs the *Gen* algorithm of the cryptosystem Π to generate a secret key *sk*. Then, he uses the *Enc* algorithm of Π with *sk* as the secret key to encrypt the set of secrets and obtains a set of m ciphertexts. Finally, he shares *sk* among participants using the sharing algorithm of Feldman's VSS scheme [38]. During *Tun*, at first, players' reputation values are renewed based on their behaviors in the past time interval. Then, using the new reputation values, the assigned weight to each participant and his shares would be updated. If an authorized subset of participants decides to reconstruct the secrets, the *Rec* algorithm of the proposed scheme is run in which, first the secret key *sk* is reconstructed. Then, using the *Dec* algorithm of the cryptosystem Π, the recovered secret key *sk* and the set of ciphertexts, the original secrets will be recovered.

We now provide the details of sharing, tuning and reconstruction algorithms in the following sections.

3.3.1 Secret Sharing (*Sha*) Algorithm

In this algorithm, the dealer performs the following steps:

(1) Runs the *Gen* algorithm of Π to get a secret key *sk*.

(2) Runs the *Enc* algorithm of Π with *sk* as the secret key m times, each time with one of the secrets $S_i (1 \leq i \leq m)$ as the plaintext to obtain a set of m ciphertexts: $\{c_1 = Enc(S_1,\ sk),\ c_2 = Enc(S_2,\ sk),\ \cdots,\ c_m = Enc(S_m,\ sk)\}$.

(3) Publishes the set of ciphertexts $\{c_1,\ c_2,\ \cdots,\ c_k\}$.

(4) Constructs the polynomial $f(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$ with $a_0 = sk$ and randomly chosen a_i from Z_q for $(1 \leq i \leq t-1)$.

(5) Computes and publishes the values $A_i = g^{a_i}\ (mod\ q)$ for $i = 0,\ \cdots,\ t-1$.

(6) Let ω_i $(i = 1,\ \cdots,\ n)$ denote the weight assigned to P_i according to his initial reputation value. Then, for each participant P_i:

   (a) For $j = 1,\ \cdots,\ \omega_i$: computes $sh_{ij} = f(x_{ij})$ where $x_{ij} = i\omega - \omega + j$ and $\omega << t$ is the maximum weight that any participant can have.

   (b) Sends $\{sh_{i1},\ \cdots,\ sh_{i\omega_i}\}$ as P_i's share from the set of secrets to him via a secure channel.

(7) Each participant P_i verifies the validity of his shares $\{sh_{ij}\}_{j=1}^{\omega_i}$ through the following relation:

$$g^{sh_{ij}} = \prod_{k=0}^{t-1} A_k^{x_{ij}^k}\ (mod\ q), \qquad (1)$$

where $x_{ij} = i\omega - \omega + j$ and broadcasts (x_{ij}, sh_{ij}) as a complaint against the dealer if Equation (1) does not hold.

(8) If the number of complaints is at least equal to t then, all participants output *reject* and stop execution of the protocol.

(9) The dealer reveals the share (sh_{ij}) corresponding to each complaining participant P_i.

(10) Each participant checks the validity of the revealed shares by the dealer through Equation (1) and outputs *reject* if any of the revealed shares fails this equation. Otherwise, outputs *accept*.

3.3.2 Social Tuning (*Tun*) Algorithm

When the set os secrets are shared, *Tun* can be executed many times. There is no need for the dealer to be online and participants can execute this algorithm by themselves. The aim of this algorithm is to enhance the authority of reliable and cooperative players by increasing their weights and reduce the authority of unreliable participants due to their past behaviors by decreasing their weights. In this algorithm, at first, participants' new reputation values are computed by a trust function such as that used in [34]. Then, based on these new values, the participants' new weights will be calculated and the shares corresponding to each participant will be renewed cooperatively by participants. Let *Arbsub* denote an arbitrary subset of participants and define $WIndex = \{(i,j) | P_i \in Arbsub \& j = 1, \cdots, \omega_i\}$. The details of this algorithm are as follows:

(1) The players, updates the reputation value and the weight assigned to each member of U by using a trust function such as that used in [34].

(2) Each player $P_i \in Arbsub$, for $j = 1, \cdots, \omega_i$:

   (a) Generates a polynomial $f^{[ij]}(x) = b_0^{[ij]} + b_1^{[ij]}x + b_2^{[ij]}x^2 + \cdots + b_{t-1}^{[ij]}x^{t-1}$ over Z_q where $b_0^{[ij]} = sh_{ij}$, and $b_1^{[ij]}, \cdots, b_{t-1}^{[ij]}$ are randomly chosen values from Z_q.

   (b) Computes $B_k^{[ij]} = g^{b_k^{[ij]}}$ for $k = 1, \cdots, t-1$ and publishes these values.

   (c) Let ω'_k denote the new weight assigned to each participant $P_k \in U$ according to his new trust value. Then, for each $P_k \in U$ and $l = 1, \cdots, \omega'_k$ computes $sh_{ij \to kl} = f^{[ij]}(x_{kl})$ as the sub-share from his j-th old share corresponding to the l-th new share of P_k and sends these values to him via a secure channel.

(3) Upon receiving his sub-shares from the set of participants executing *Tun*, each player $P_k \in U$

computes the value $B_0^{[ij]} = \prod_{k=0}^{t-1} A_k^{x_{[ij]}^k} \ (mod\ q)$ for each of his received sub-shares $(sh_{ij \to lm})$. Then, checks the validity of each of his sub-shares through the following relation:

$$g^{sh_{ij \to kl}} = \prod_{v=0}^{t-1} {B_v^{[ij]}}^{x_{kl}^v} \ (mod\ q), \qquad (2)$$

where i and j is such that $P_i \in Arbsub$ and $P_k \in U$, respectively, $j = 1, \cdots, \omega_i$, and $l = 1, \cdots, \omega'_k$. P_k broadcasts (ij, kl) as a complaint if the equation does not hold for $sh_{ij \to kl}$.

(4) The participants remove (i, j) from $WIndex$ if there is at least t different complaints against j-th polynomial of P_i (*i.e.*, t complaints with $(ij, **)$ form).

(5) Each $P_i \in U$, for each $j = 1, \cdots, \omega_i$, publishes the value $sh_{ij \to kl}$ corresponding to each broadcasted complaint.

(6) Participants check the validity of the published value using Equation (2) and remove (i, j) from $WIndex$ if it does not hold for at least one of the published values from j-th polynomial of P_i.

(7) Let $|WIndex|$ denote the size of the set $WIndex$. Then, if $|WIndex| \geq t$, each player $P_k \in U$ erases his old shares from the set of secrets and computes his new shares as $sh_{kl} = \sum_{i:(i,*) \in WIndex} \sum_{j:(i,j) \in WIndex} sh_{ij \to kl} l_{ij}(0)$ for $l = 1, \cdots, \omega'_l$ where $*$ denotes an arbitrary index and $l_{ij}(\cdot)$ denotes the corresponding Lagrange polynomial to the pair (i, j) from $WIndex$.

(8) Each player $P_k \in U$ updates the public values A_k for $k = 1, \cdots, t-1$ as $A_k = \prod_{i:(i,*) \in WIndex} \prod_{j:(i,j) \in WIndex} {B_k^{[ij]}}^{l_{ij}(0)}$.

3.3.3 Reconstruction (*Rec*) Algorithm

Let assume that *Arbsub* be an arbitrary subset of participants. Then, a trusted party (called the share combiner) can perform the following steps to reconstruct the set of secrets:

(1) Checks the validity of each of the provided shares through the following relation:

$$g^{sh_{ij}} = \prod_{k=0}^{t-1} A_k^{x_{ij}^k} \ (mod\ p), \qquad (3)$$

where i is such that $P_i \in Arbsub$ and $j = 1, \cdots, \omega_i$.

(2) If the number of valid shares (those for which the above equation holds) is at least equal to t, uses Lagrange interpolation method to compute the secret key sk.

(3) Reconstructs the set of original secrets as $\{S_1 = Dec(c_1,\ sk),\ S_2 = Dec(c_2,\ sk),\ \cdots, S_m = Dec(c_m,\ sk)\}$.

4 Security and Performance Analysis

In this section, we first prove that the proposed SMSS scheme achieves computational security in the presence of active mobile adversaries. Then, the proposed scheme is compared with the only existing SSS scheme secure against active mobile adversaries (*i.e.*, Nojoumian et al.'s second scheme) in terms of communication complexity, computational complexity, the share size, the number of public parameters, and the achieved security and features.

4.1 Security Analysis

In this section, we prove that our proposed SMSS scheme is strongly secure under active mobile adversary model. In order to do so, we first state the following lemmas which prove that *Sha*, *Tun* and *Rec* algorithms of the proposed scheme are computationally secure under an active adversary model.

Lemma 1. *The Tun algorithm of the proposed scheme is computationally secure,* i.e., *non-authorized subsets of participants obtain no information about the secret key sk from their shares through this algorithm.*

Proof. The *Tun* algorithm of the proposed SMSS scheme can be considered as follows. At first, each participant distributes his shares by using Feldman's VSS scheme. Then, participants erase their old shares and compute their new shares as a linear combination of the received shares from all the participants executing this algorithm. The computational security of Feldman's VSS scheme makes it computationally infeasible for non-authorized subsets of participants to obtain information about the old shares of participants executing this algorithm. Therefore, the only information that the adversary has access to is the set of shares corresponding to a non-authorized subset of participants from different time intervals. Since the shares from different time intervals are obtained from different polynomials, they can not be used together to obtain information on the secret key sk. Therefore, this algorithm does not give any advantage to the adversary compared to what he has from the *Sha* and *Rec* algorithms. □

Lemma 2. *In the proposed scheme, no non-authorized subset of participants can obtain information about the secret key sk that is used to encrypt the set of original secrets.*

Proof. In the proposed scheme, a KPA-secure encryption scheme is used to encrypt the set of original secrets. KPA-security of the used cryptosystem makes it computationally impossible for a polynomially bounded adversary to obtain information about the the secret key sk from the set of ciphertexts $c_1, \cdots, c_m$. Without obtaining any information from the ciphertexts, using the shares is the only remaining way in which the adversary can obtain information on the secret. In the following, it is shown that obtaining information about the secret is computationally infeasible assuming that an adversary can gain access to shares of only non-authorized subsets of participants in each time interval. The *Sha* and *Rec* algorithms of the proposed scheme are exactly the same as those in Feldman's VSS scheme which indicates that these algorithms of the proposed scheme are computationally secure. The *Tun* algorithm is what makes the proposed scheme different from Feldman's scheme and its computational security is shown in Lemma 1. Security of all the three algorithms indicates the computational security of sk in the proposed scheme. □

Theorem 1. *The proposed social multi-secret sharing scheme is strongly secure under active mobile adversary model.*

Proof. Assume that $m-1$ of the secrets shared by the proposed scheme are revealed in clear (note that the worst case is considered) and S_m is the only secret about which no information is known. At first, note that Lemma 2 states that it is computationally infeasible for non-authorized subsets of participants to obtain information about the secret key sk. Now, KPA-security of the used cryptosystem makes it computationally impossible for a polynomially bounded adversary to obtain information about other ciphertexts (*e.g.*, c_m) by using the known (plaintext, ciphertext) pairs. Therefore, it can be concluded that the proposed scheme is strongly secure in the active mobile adversary model. □

4.2 Comparison

In this section, the proposed scheme (denoted here by SMSSS) is compared with Nojoumian et al.'s second scheme (denoted by SSSS) which, the same as the proposed scheme is secure under active mobile adversary model. The comparisons is done in terms of communication and computational complexity, the share size, the number of public parameters, the security provided by each of the schemes and the properties that each of the schemes achieve. The results of the comparisons are summarized in Table 1.

4.2.1 Communication Complexity

In this section, the number of communication rounds required in each algorithm of the schemes is compared:

- **The *Sha* algorithm:** This algorithm of SSSS requires two communication rounds, one for the share distribution by the dealer and another one for the pairwise verification of shares between participants. SMSSS requires one communication round in the *Sha* algorithm. Note that no communication is needed for the share verification in SMSSS.
- **The *Tun* algorithm:** This algorithm of SSSS requires three communication rounds (one round in phase I of its *Tun* algorithm and two more communication rounds in phase II of that algorithm, please see [34]). SMSSS requires one communication round in the *Tun* algorithm in which the share of each participant is transferred to him.
- **The *Rec* algorithm:** Both of the schemes requires one communication round in their *Rec* algorithm to send the shares to the share combiner.

Based on these statements, it can be concluded that SMSSS outperforms SSSS in terms of the communication complexity. The results are summarized in Table 1.

4.2.2 Computational Complexity

Let n denote the maximum number of parties who can join the scheme and let t be the threshold parameter of the schemes; note that $n > t$. Also, let w (for the sake of simplicity we assign $w = t$) be the maximum weight of each player in the schemes. Let $\mathcal{M}u$, $\mathcal{E}n$, $\mathcal{D}e$, and $\mathcal{E}x$ denote a multiplication, a symmetric encryption, a symmetric decryption, and an exponentiation operation, respectively. The operations $\mathcal{E}n$, $\mathcal{D}e$ and $\mathcal{E}x$ can be done by performing some fixed number of $\mathcal{M}u$ operations. Therefore, the number of operations $\mathcal{E}n$, $\mathcal{D}e$ and $\mathcal{E}x$ can be approximated by $c_1\mathcal{M}u$, $c_2\mathcal{M}u$, $c_3\mathcal{M}u$ for some constants c_1, c_2 and c_3, respectively. In the following, an approximate computational comparison between the SMSSS and SSSS is brought.

In *Sha* algorithm of SMSSS, at first, the dealer needs to encrypt m messages. This process can be done by m $\mathcal{E}n$ operations. Then, he should compute the shares that correspond to each participant $P_i \in U$. Each share generation can be done via evaluation of a $(t-1)$-th degree polynomial on a random input which can be done by $O(t)$ $\mathcal{M}u$ operations. Therefore, generating the shares corresponding to P_i can be done by $\omega_i t$ $\mathcal{M}u$ operations. Hence, the overall complexity of this process is $O(nt^2)\mathcal{M}u$ operations. Finally, the dealer should perform t exponentiations to generate the required public values for verification. Therefore, the overall computations that the dealer needs to perform is $O(m + t^2 n)\mathcal{M}u$ operations. Moreover, in *Sha* algorithm of SMSSS, each participant $P_i \in U$ needs to verify the validity of each of the received shares. Each

Table 1. Comparisons Between SMSSS and SSSS with n as the Number of Participants, t as the Threshold Parameter and m as the Number of Secrets in the Proposed Scheme.

Scheme			SSSS	SMSSS
Computational complexity (# *of* $\mathcal{M}u$ operations)	Sha	Each participant	$O(nt^3)$	$O(t^2)$
		The dealer	$O(nt^2)$	$O(m+nt^2)$
	Tun	Each participant	$O(n^2t^4)$	$O(nt^3)$
	Rec	The share combiner	$O(t^4)$	$O(m+t^2)$
Communication complexity	Sha		2	1
	Tun		3	1
	Rec		1	1
The share size			$O(t^2\|q\|)$	$O(t\|q\|)$
Achieved security level			Unconditional	Computational
Multi-secret sharing			No	Yes
# of public parameters			0	$O(t^2)$

share verification requires t $\mathcal{M}u$ and t $\mathcal{E}x$ operations. Therefore, the overall computations needed to be done by each participant in *Sha* algorithm of SMSSS can be approximated as $O(t^2)$ $\mathcal{M}u$ operations. To generate each share, in SSSS, the dealer computes a univariate polynomial of degree $t-1$ as a share by computing the value of a bivariate polynomial of degree $t-1$ at a point with fixed first coordinate. This can be done by $(t-1)$ $\mathcal{M}u$ operations. Therefore, generating the shares corresponding to P_i can be done by $\omega_i t$ $\mathcal{M}u$ operations. Hence, the overall complexity of this process is $O(nt^2)\mathcal{M}u$ operations. To verify the validity of each received share in SSSS, each participant, for each of the received shares does a pairwise checking with all the shares of all participants (including himself). Each pairwise checking is done by computing the value of a polynomial of degree $t-1$ at a point. As explained earlier this process can be done in $O(t)$ $\mathcal{M}u$ operations. The number of pairwise checking for each share is $\sum_{i=1}^{n} \omega_i - 1$. Therefore, the overall computations that is required to be done by each participant to verify the validity of all of his received shares is $O(nt^3)$ $\mathcal{M}u$ operations.

In *Tun* algorithm of SMSSS, each participant, as a dealer, shares each one of his old shares. Moreover, the participants need to verify the validity of their received shares. Based on the computations done in the previous paragraph, the computations needed to be done by each of the participants in this algorithm is $O(nt^3)$ $\mathcal{M}u$ operations. Phase II of SSSS is computationally more complex than phase I of their algorithm. That is because the verifications are done in the second phase. In this phase, each participant for each of his shares acts as a dealer and shares the value zero among the participants. Using the notations brought at the beginning of this section, to accomplish this goal, the computations that is needed to be carried out by each participant is $O(nt^3)$ $\mathcal{M}u$ operations. Moreover, the participants require to verify the validity of their received shares by pairwise checking. To do this process, each participant needs to carry out $O(n^2t^4)$ $\mathcal{M}u$ operations. Therefore, the overall computations that is needed to be done by each participant in *Tun* algorithm of SSSS is $O(n^2t^4)$ $\mathcal{M}u$ operations.

In the *Rec* algorithm of SMSSS, the share combiner needs to verify the validity of the provided shares and then reconstruct the secret key sk using lagrange interpolation method. Finally, he uses the secret key sk to decrypt the set of ciphertexts. Based on the computational complexity of the share verification of SMSSS, that of Lagrange interpolation method and decryption operations, the overall computations that is needed to be done by the share combiner in this algorithm of SMSSS is $O(t^2+m)$ $\mathcal{M}u$ operations. In the *Rec* algorithm of SSSS, same as in SMSSS, at first, the share combiner needs to verify the validity of the received shares by pairwise checking for each two of the received shares. Then, he needs to reconstruct the secret using the Lagrange interpolation method. Therefore, the overall computations that is needed to be done by the share combiner in *Rec* algorithm of SSSS is $O(t^4)$ $\mathcal{M}u$ operations.

Based on the achieved computational complexities, it can be concluded that the proposed scheme (which is able to share multiple secrets) outperforms Nojoumian et al.'s scheme in terms of computational complexity (which is able to share only one secret).

4.2.3 The Share Size

In SSSS, each participant P_i receives ω_i shares where ω_i is the weight assigned to him and each share's size is equal to $t|q|$ where t is the threshold parameter and $|q|$ is the bit length of the prime number q. Therefore, the total share size of each participant P_i is equal to $t\omega_i|q| = O(t^2|q|)$. In SMSSS, each participant P_i receives ω_i shares, each of them of the size $|q|$. Therefore, the total share size of each participant P_i in SMSSS is equal to $\omega_i|q| = O(t|q|)$. From this statement, it can be seen that the share size of each participant in the proposed scheme is $\frac{1}{t}$-th of that in Nojoumian et al.'s scheme.

4.2.4 Security

While SSSS is unconditionally secure under active mobile adversary model, SMSSS is computationally secure in the same model. Therefore, SSSS guarantees a more strong security definition.

4.2.5 Achieved Properties

SSSS is a verifiable social secret sharing scheme which can only share one secret during each execution of it. SMSSS, while preserving all the properties achieved by SSSS, is a multi-secret sharing scheme.

4.2.6 Public Parameters

In *Sha* algorithm of SMSSS, in order to make the participants capable of verifying the validity of their shares, the dealer publishes t public values from Z_q. Moreover, in the *Tun* algorithm of SMSSS, each participant P_i needs to publish $\omega_i(t-1)$ public values. Therefore, the maximum number of public parameters in SMSSS is $O(t^2)$. Note that in SMSSS, after each execution of *Tun*, the number of public values would be reduced to t. Since SSSS achieves the verifiability property by using bivariate symmetric polynomials, it does not require to publish any public parameters. Therefore, Nojoumian et al.'s scheme outperforms the proposed one in terms of the number of public parameters.

5 Conclusion

In this paper, the concept of social secret sharing (SSS) is generalized to social multi-secret sharing (SMSS). Then, by combining a modified version of Nojoumian et al.'s SSS scheme and symmetric encryption schemes, a construction for SMSS scheme is proposed. Afterward, it is proved that the proposed scheme is strongly secure in active adversarial model. Finally, the efficiency of the proposed scheme is demonstrated by comparing it with the only existing SSS scheme with provable security in active adversarial model.

References

[1] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, November 1979. ISSN 0001-0782. doi:10.1145/359168.359176.

[2] George Robert Blakley. Safeguarding cryptographic keys. In *Proceedings of the national computer conference*, volume 48, pages 313–317, 1979.

[3] Wen-Ai Jackson, Keith M. Martin, and Christine M. O'Keefe. Multisecret Threshold Schemes. In *Advances in Cryptology: Crypto '93*, volume 773 of *Lecture Notes in Computer Science*, pages 126–135, 1994. doi:10.1007/3-540-48329-2_11.

[4] Z. Eslami, N. Pakniat, and M. Noroozi. Hierarchical threshold multi-secret sharing scheme based on Birkhoff interpolation and cellular automata. In *Computer Architecture and Digital Systems (CADS), 2015 18th CSI International Symposium on*, pages 1–6, Oct 2015.

[5] Liao-Jun Pang and Yu-Min Wang. A new (t,n) multi-secret sharing scheme based on Shamir's secret sharing. *Applied Mathematics and Computation*, 167(2):840 – 848, 2005. doi:10.1016/j.amc.2004.06.120.

[6] Massoud Hadian Dehkordi and Samaneh Mashhadi. An efficient threshold verifiable multi-secret sharing. *Computer Standards and Interfaces*, 30 (3):187 – 190, 2008. doi:10.1016/j.csi.2007.08.004.

[7] Massoud Hadian Dehkordi and Samaneh Mashhadi. New efficient and practical verifiable multi-secret sharing schemes. *Information Sciences*, 178(9):2262 – 2274, 2008. doi:10.1016/j.ins.2007.11.031.

[8] Ziba Eslami and Saideh Kabiri Rad. A New Verifiable Multi-secret Sharing Scheme Based on Bilinear Maps. *Wireless Personal Communications*, 63(2):459–467, 2012. ISSN 1572-834X. doi:10.1007/s11277-010-0143-0.

[9] Shyamalendu Kandar and Bibhas Chandra Dhara. A (k, n) multi secret sharing scheme using two variable one way function with less public values. In Sushil Jajodia and Chandan Mazumdar, editors, *Information Systems Security: 11th International Conference, ICISS 2015, Kolkata, India, December 16-20, 2015. Proceedings*, pages 532–541, 2015.

[10] Zhenhua Chen, Shundong Li, Youwen Zhu, Jianhua Yan, and Xinli Xu. A cheater identifiable multi-secret sharing scheme based on the Chinese remainder theorem. *Security and Communication Networks*, 8(18):3592–3601, 2015.

[11] Jingmin He and Edward Dawson. Multistage

secret sharing based on one-way function. *Electronics Letters*, 30(19):1591–1592, 1994.

[12] H. Pilaram and T. Eghlidos. An Efficient Lattice Based Multi-Stage Secret Sharing Scheme. *IEEE Transactions on Dependable and Secure Computing*, 14(1):2–8, Jan 2017. ISSN 1545-5971. doi:10.1109/TDSC.2015.2432800.

[13] Mitra Fatemi, Reza Ghasemi, Taraneh Eghlidos, and Mohammad Reza Aref. Efficient multistage secret sharing scheme using bilinear map. *IET Information Security*, 8:224–229(5), 2014. doi:10.1049/iet-ifs.2013.0046.

[14] Samaneh Mashhadi. New multi-stage secret sharing in the standard model. *Information Processing Letters*, 127:43 – 48, 2017. ISSN 0020-0190. doi:10.1016/j.ipl.2017.07.002.

[15] Samaneh Mashhadi, Massoud Hadian Dehkordi, and Niloofar Kiamari. Provably secure verifiable multi-stage secret sharing scheme based on monotone span program. *IET Information Security*, 11: 326–331(5), 2017. doi:10.1049/iet-ifs.2017.0111.

[16] Hung-Yu Chien, JAN Jinn-Ke, and Yuh-Min Tseng. A practical (t, n) multi-secret sharing scheme. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 83(12):2762–2765, 2000.

[17] Chou-Chen Yang, Ting-Yi Chang, and Min-Shiang Hwang. A (t,n) multi-secret sharing scheme. *Applied Mathematics and Computation*, 151(2):483 – 490, 2004. doi:10.1016/S0096-3003(03)00355-2.

[18] Jun Shao and Zhenfu Cao. A new efficient (t,n) verifiable multi-secret sharing (VMSS) based on YCH scheme. *Applied Mathematics and Computation*, 168(1):135 – 140, 2005. ISSN 0096-3003. doi:10.1016/j.amc.2004.08.023.

[19] Nasrollah Pakniat, Mahnaz Noroozi, and Ziba Eslami. Reducing Multi-Secret Sharing Problem to Sharing a Single Secret Based on Cellular Automata. *Journal on Computer Science and Engineering*, 14(1):38 – 43, 2016.

[20] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk, and Moti Yung. Proactive secret sharing or: How to cope with perpetual leakage. In *Advances in Cryptology — CRYPT0' 95*, pages 339–352. Springer, 1995. doi:10.1007/3-540-44750-4_27.

[21] Christian Cachin, Klaus Kursawe, Anna Lysyanskaya, and Reto Strobl. Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 88–97. ACM, 2002. doi:10.1145/586110.586124.

[22] Lidong Zhou, Fred B. Schneider, and Robbert Van Renesse. APSS: Proactive Secret Sharing in Asynchronous Systems. *ACM Transactions on Information and System Security (TISSEC)*, 8 (3):259–286, 2005. doi:10.1145/1085126.1085127.

[23] Douglas R Stinson and Ruizhong Wei. Unconditionally secure proactive secret sharing scheme with combinatorial structures. In *Selected areas in cryptography*, volume 1758, pages 200–214. Springer, 2000. doi:10.1007/3-540-46513-8_15.

[24] David A. Schultz, Barbara Liskov, and Moses Liskov. Mobile Proactive Secret Sharing. In *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*, PODC '08, pages 458–458. ACM, 2008. doi:10.1145/1400751.1400856.

[25] Amos Beimel, Tamir Tassa, and Enav Weinreb. Characterizing Ideal Weighted Threshold Secret Sharing. In Joe Kilian, editor, *Theory of Cryptography*, pages 600–619. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-30576-7. doi:10.1007/978-3-540-30576-7_32.

[26] Ernest F. Brickell. Some ideal secret sharing schemes. In *Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*, EUROCRYPT '89, pages 468–475, 1990.

[27] Gustavus J. Simmons. How to (Really) Share a Secret. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 390–448. Springer New York, 1990. ISBN 978-0-387-34799-8. doi:10.1007/0-387-34799-2_30.

[28] Javier Herranz and Germán Sáez. New results on multipartite access structures. *IEE Proceedings - Information Security*, 153(4):153–162, 2006.

[29] Tamir Tassa. Hierarchical Threshold Secret Sharing. *Journal of Cryptology*, 20(2):237–264, 2007.

[30] Ching-Fang Hsu and Lein Harn. Multipartite Secret Sharing Based on CRT. *Wireless Personal Communications*, 78(1):271–282, 2014. doi:10.1007/s11277-014-1751-x.

[31] Nasrollah Pakniat, Mahnaz Noroozi, and Ziba Eslami. Distributed key generation protocol with hierarchical threshold access structure. *IET Information Security*, 9(4):248 – 255, 2015.

[32] Ryszard Smarzewski and Joanna Kapusta. Algorithms for multi-secret hierarchical sharing schemes of Shamir type. *Annales Universitatis Mariae Curie-Sklodowska, sectio AI–Informatica*, 3(1):65–91, 2015.

[33] Xukai Zou, Fabio Maino, Elisa Bertino, Yan Sui, Kai Wang, and Feng Li. A New Approach to Weighted Multi-Secret Sharing. In *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, 2011. doi:10.1109/ICCCN.2011.6005766.

[34] Mehrdad Nojoumian, Douglas R Stinson, and Morgan Grainger. Unconditionally secure social secret sharing scheme. *IET information security*, 4(4):202–211, 2010.

[35] Ziba Eslami, Nasrollah Pakniat, and Mehrdad Nojoumian. Ideal social secret sharing using Birkhoff interpolation method. *Security and Communication Networks*, 9(18):4973–4982, 2016. doi:10.1002/sec.1668.

[36] Javier Herranz, Alexandre Ruiz, and Germán Sáez. New results and applications for multi-secret sharing schemes. *Designs, Codes and Cryptography*, 73(3):841–864, 2014. doi:10.1007/s10623-013-9831-6.

[37] Carlo Blundo, Alfredo De Santis, Giovanni Di Crescenzo, Antonio Giorgio Gaggia, and Ugo Vaccaro. Multi-Secret Sharing Schemes. In *Advances in Cryptology — CRYPTO '94*, pages 150–163. Springer Berlin Heidelberg, 1994. doi:10.1007/3-540-48658-5_17.

[38] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 427–438. IEEE, 1987. doi:10.1109/SFCS.1987.4.

[39] Mehrdad Nojoumian and Timothy C Lethbridge. A new approach for the trust calculation in social networks. In *E-Business and Telecommunication Networks*, pages 64–77. Springer, 2008. doi:10.1007/978-3-540-70760-8_6.

Nasrollah Pakniat received his B.S. degree in Computer Science in 2008 from Shahid Bahonar University, Kerman, Iran. In 2011, he received his M.S. degree in Computer Science from Shahid Beheshti University, Tehran, Iran. He received his Ph.D. degree in 2015 in Mathematics from Shahid Beheshti University. He is currently an assistant professor at Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Iran. His research interests include applied cryptography, network security and text mining.

Ziba Eslami received her B.S., M.S., and Ph.D. in Applied Mathematics from Tehran University in Iran. She received her Ph.D. in 2000. From 1991 to 2000, she was a resident researcher in the Institute for Studies in Theoretical Physics and Mathematics (IPM), Iran. During the academic years 2000-2003, she was a Post Doctoral Fellow in IPM. She served as a non-resident researcher at IPM during 2003-2005. Currently, she is an associate professor in the Department of Computer Science at Shahid Beheshti University in Iran. Her research interests include design theory, combinatorial algorithms, cryptographic protocols, and steganography.