The ISC Int'l Journal of Information Security

ISeCure

July 2015, Volume 7, Number 2 (pp. 91–99)

http://www.isecure-journal.org

Computationally Secure Multiple Secret Sharing: Models, Schemes, and Formal Security Analysis

Samaneh Mashhadi [1,*]

[1] *Department of Mathematics, Iran University of Science & Technology, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 2 June 2015
Revised: 30 August 2015
Accepted: 14 October 2015
Published Online: 20 October 2015

Keywords:
Multi-secret Sharing Scheme, Multi-stage Secret Sharing Scheme, Provable Security, Private-key Cryptosystem, Standard Model.

ABSTRACT

A multi-secret sharing scheme (MSS) allows a dealer to share multiple secrets among a set of participants. in such a way a multi-secret sharing scheme (MSS) allows a dealer to share multiple secrets among a set of participants, such that any authorized subset of participants can reconstruct the secrets. Up to now, existing MSSs either require too long shares for participants to be perfect secure, or do not have a formal security analysis/proof. In 2013, Herranz *et al.* provided the first formal definition of computational security for multi-stage secret sharing scheme (MSSS) in the standard model and proposed a practical and secure scheme. As far as we know, their scheme is the only computationally secure MSS in the standard model, and there is no formal definition of the computational security for other categories of MSSs. Based on this motivation, in this paper, we define the first formal model of indistinguishability against the chosen secret attacks (CSA) for other types of MSSs in the standard model. Furthermore, we present two practical CSA-secure MSSs, belonging to different types of MSSs and enjoying the advantage of short shares. They are also provably secure in the standard model. Based on the semantic security of the underlying encryption schemes, we prove the security of our schemes.

1 Introduction

A *secret sharing scheme* is a randomized protocol for the distribution of a secret s among n participants $\mathcal{P} = \{P_1, \ldots, P_n\}$ according to some access structures $\Gamma \subseteq 2^{\mathcal{P}}$ such that any authorized subset of participants can reconstruct the secret value by putting their shares together, but any unauthorized subset participants cannot get any information about the secret s.

In the literature we deal with two different notations of secrecy according to the meaning of "any information" in the above formulation. One is *perfect secrecy* and the other is *computational secrecy*. In a *perfect secret sharing scheme* [19] attackers have unlimited computational resources and the security of the scheme is not based on cryptographic assumptions; while, in a *computational secret sharing scheme* [14], attackers are modeled as polynomial-time algorithms and the security of scheme depends on some computational assumptions. Unfortunately, perfect secret sharing schemes suffer from severe lower bounds on the shares length; namely the share of each participant must be, at least, as long as the length of the secret. Obviously, this is inefficient when the secret is a big privacy file, a large message transmitted on an insecure channel or enormous data in distributed storage. In order to make up for this deficiency, Krawczyk [14] presented a computational secret sharing scheme.

* Corresponding author.
Email address: smashhadi@iust.ac.ir (S. Mashhadi)
ISSN: 2008-2045

Multi-secret sharing scheme is a generalization of secret sharing scheme and in the real world applications, multi-secret sharing schemes are very practical. In a multi-secret sharing scheme, multiple secrets are distributed among the participants during a secret sharing process. Two categories of Multi-secret sharing scheme according to the secret reconstruction is proposed, the multi-stage and the general multi-secret sharing scheme; and depending on any specific situation, each category may be preferable. In a *general multi-secret sharing scheme* (GMSS), all of the secrets are reconstructed simultaneously in one stage [6, 20]; while, in a *multi-stage secret sharing scheme* (MSSS), the secrets have different levels of importance, and any authorized subset of participants can recover only one secret in every stage [5, 8, 9, 15]. In the literature, there are two different types of MSSSs. In the first type (MSSST1), the secret reconstruction can be executed in any order, e.g. the schemes [8, 9]. In the second type (MSSST2), the secret reconstruction must be executed in a predefined order [5, 15] are examples of this type.

Most of the works on multi-secret sharing schemes have focused on *perfect secure* multi-secret sharing schemes. Perfect secure multi-secret sharing schemes, similar to perfect secret sharing schemes, have very long shares and this makes perfect secure multi-secret sharing schemes impractical; namely the size of each share should be at least equal to the sum of the size of different secrets [18]. To overcome the drawback of perfect multi-secret sharing schemes, several works on *computational secure* multi-secret sharing schemes is recently done; but unfortunately, the authors of these schemes introduced new construction of computational secure multi-secret sharing schemes without providing formal proofs of the proposed schemes [4, 7, 8, 10, 16, 17, 20]. Until, in 2013, Herranz *et al.* [9] proposed the first computational multi-stage secret sharing scheme in the standard model, which was extension of some previous works [8, 16] and belonged to the MSSST1. They provided the formal definition of security for this type in the standard model. Moreover, they proved that if a *chosen plaintext attack* (CPA) secure encryption scheme is used in construction of their MSSST1, this scheme has *indistinguishability against chosen secret attacks* (CSAs) in the standard model.

To the best of our knowledge, except Herranz *et al.*'scheme, computational multi-secret sharing schemes in the standard model have not been treated in the literature. Our current work aims to fill this gap. Our goal is to formally prove the computational security of the other types of multi-secret sharing schemes in the standard model. The main contributions of this paper is as follows. A brief review of some preliminaries required throughout the paper is described at first. Then, we provide the formal definitions of indistinguishability against the CSAs for the computational MSSST2 and computational GMSS, respectively. After describing formally the computational security of a MSSST2, we present an efficient CSA-secure MSSST2 (it can be thought as a generalization of some previous MSSSs [8] and [16]) and prove its security in the standard model. The proposed scheme enjoys the same level of security as Herranz *et al.*'s scheme. Furthermore, we propose a practical CSA-secure GMSS which is inspired by previous work in this area [14] and based on the semantic security of the underlying encryption scheme, we prove its security against chosen secret attacks in the standard model. Finally, we give a performance comparison of our schemes with the Herranz *et al.*'s scheme. The most important part of this work is the formal security analysis that we provide, for both proposed multi-secret sharing schemes.

2 Formal Definitions of Secure Private Key Encryption

In the following, we review the definitions of the M.Eav-secure and CPA-secure private key encryption schemes and the game-based security definition models [1–3, 12, 13].

2.1 Private-key Encryption Scheme

A private-key encryption scheme is a tuple of $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that:

- The randomized key-generation algorithm Gen takes as input a security parameter 1^λ and outputs a random key k; $k \leftarrow \mathsf{Gen}(1^\lambda)$.
- The randomized encryption algorithm Enc takes as input a key k and a plaintext message m, and outputs a random ciphertext c; $c \leftarrow \mathsf{Enc}(k, m)$.
- The deterministic decryption algorithm Dec takes as input a key k and a ciphertext c, and outputs a message m, such that $m := \mathsf{Dec}(k, c)$.

For correctness, $\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$ must hold, for any k, m.

2.2 The Multiple-message Eavesdropping Indistinguishability Experiment

Security of a private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ under eavesdropper attacks, is defined by the following game between an adversary $\mathcal{A}_1$ and a challenger.

Game $\mathcal{G}_1$

(1) The challenger chooses a random bit $\beta \in \{0, 1\}$ and runs $\mathsf{Gen}(1^\lambda)$, to produce a secret key k.

(2) $\mathcal{A}_1$ chooses tuples $M_0 = (m_1^0, \ldots, m_t^0)$ and $M_1 = (m_1^1, \ldots, m_t^1)$ where $|m_i^0| = |m_i^1|$ for all $i = 1, \ldots, t$, and $q_c = t$ is the number of challenges.

(3) The challenger runs $c_i \leftarrow \mathsf{Enc}(k, m_i^\beta)$ and sends back $(c_1, \ldots, c_t)$ to $\mathcal{A}_1$.

(4) $\mathcal{A}_1$ outputs a bit β'.

(5) The output of the game is defined to be 1 if $\beta = \beta'$ and 0 otherwise. We write $\mathrm{PrivK}_{\mathcal{A}_1}^{\mathrm{M.Eav}}(\lambda) = 1$ if the output is 1 and in this case we say that $\mathcal{A}_1$ succeeded.

A private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has indistinguishable multiple encryptions in the presence of an eavesdropper (or is M.Eav-secure) in the computational scenario, if for any polynomial-time q_c-adversary $\mathcal{A}_1$ there exists a negligible function negl such that

$$\Pr[\mathrm{PrivK}_{\mathcal{A}_1}^{\mathrm{M.Eav}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

2.3 The Chosen-plaintext Attacks Indistinguishability Experiment

Security of a private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ under chosen-plaintext attack (CPA), in the multi-user setting, is defined by the following game between an adversary $\mathcal{A}_2$ and a challenger.

Game $\mathcal{G}_2$

(1) The challenger chooses a random bit $\beta \in \{0, 1\}$.

(2) $\mathcal{A}_2$ chooses the number κ of keys in the game.

(3) The challenger runs κ times $\mathsf{Gen}(1^\lambda)$, to produce κ secret keys $k_1, \ldots, k_\kappa$.

(4) $\mathcal{A}_2$ can make, at any time, q_e encryption queries (i, m) of its choice, where $i \in \{1, \ldots, \kappa\}$. As the answer, $\mathcal{A}_2$ receives the ciphertext $c \leftarrow \mathsf{Enc}(k_i, m)$.

(5) $\mathcal{A}_2$ chooses tuples (i_j, m_j^0, m_j^1), and sends these queries to challenger, where $i_j \in \{1, \ldots, \kappa\}$ and $m_j^0 \neq m_j^1$ have the same length, for all $j = 1, \ldots, q_c$, and q_c is the number of challenges.

(6) The challenger runs $c_j^* \leftarrow \mathsf{Enc}(k_{i_j}, m_j^\beta)$ and sends back to $\mathcal{A}_2$ for $j = 1, \ldots, q_c$.

(7) $\mathcal{A}_2$ outputs a bit β'.

(8) The output of the game is defined to be 1 if $\beta = \beta'$ and 0 otherwise. We write $\mathrm{PrivK}_{\mathcal{A}_2}^{\mathrm{CPA}}(\lambda) = 1$ if the output is 1 and in this case we say that $\mathcal{A}_2$ succeeded.

A private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has indistinguishable encryptions under a chosen-plaintext attack (or is CPA-secure) in the computational scenario, if for any polynomial-time (κ, q_e, q_c)-adversary $\mathcal{A}_2$ there exists a negligible function negl such that

$$\Pr[\mathrm{PrivK}_{\mathcal{A}_2}^{\mathrm{CPA}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

3 Computational Multi-secret Sharing Schemes

In this section, we define the formal models of different categories of computational multi secret sharing schemes.

3.1 Formal Model of Computational MSSST2

In a MSSST2 the dealer wants to share l secrets $s_1, s_2, \ldots, s_l$, according to l access structures $\Gamma_1, \ldots, \Gamma_l$, respectively (such that $\Gamma_i \subseteq \Gamma_{i-1}$, for $i = 2, \ldots, l$). The secrets are reconstructed stage-by-stage in special order $s_1, s_2, \ldots, s_l$. In the following, we will propose the definition of a CSA-secure computational MSSST2 and the game-based security definition model.

3.1.1 Multi-stage Secret Sharing Schemes

A computational MSSST2 is a tuple of $\Omega_1 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$ such that:

- The setup algorithm Stp takes as input a security parameter 1^λ, the set of participants $\mathcal{P}$ and the l different level access structures $\Gamma_1, \ldots, \Gamma_l$, such that $\Gamma_i \subseteq \Gamma_{i-1}$, for $i = 2, \ldots, l$ and outputs some public and common parameters pms for the scheme; $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, \{\Gamma_j\}_{1 \leq j \leq l})$.
- The distribution algorithm Dist takes as input pms and the global secret $\mathsf{s} = (s_1, \ldots, s_l)$ to be shared, and generates the set of secret shares $\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}$ and possibly some pubic output $\mathsf{out}_{\mathsf{pub}}$; $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out}_{\mathsf{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathsf{s})$.
- The reconstruction algorithm Rec takes as input $\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}$, an index $j \in \{1, \ldots, l\}$, a possible value s'_{j-1} for the $(j-1)-$th secret; and the shares $\{\mathsf{sh}_i\}_{P_i \in A}$ of the participants in some subset $A \subset \mathcal{P}$ and outputs a possible value s'_j for the $j-$th secret; $s'_j := \mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, j, s'_{j-1}, \{\mathsf{sh}_i\}_{P_i \in A})$.

For correctness, we require that for any index $j \in \{1, \ldots, l\}$ and any subset $A \in \Gamma_j$, it holds $s_j = \mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, j, s_{j-1}, \{\mathsf{sh}_i\}_{P_i \in A})$.

3.1.2 Chosen Secret Attack Indistinguishability Experiment

We now define a game for any multi-stage secret sharing scheme $\Omega_1 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$, between an

adversary $\mathcal{A}_3$ and a challenger.

Game $\mathcal{G}_3$

(1) The challenger chooses a random bit $b \in \{0,1\}$.

(2) $\mathcal{A}_3$ publishes the set of participants $\mathcal{P}$ and l access structures $\Gamma_1, \ldots, \Gamma_l \subset 2^{\mathcal{P}}$ s.t. $\Gamma_i \subseteq \Gamma_{i-1}, \forall 2 \leq i \leq l$.

(3) The challenger runs $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, \{\Gamma_j\}_{1 \leq j \leq l})$ and sends pms to $\mathcal{A}_3$.

(4) $\mathcal{A}_3$ broadcasts a subset $B \subset \mathcal{P}$ of corrupted participants.

(5) $\mathcal{A}_3$ broadcasts two different global secrets $\mathbf{s}^0 = (s_1^0, \ldots, s_l^0) \neq (s_1^1, \ldots, s_l^1) = \mathbf{s}^1$ with the following restriction: $s_j^0 = s_j^1$ for all j s.t. $B \in \Gamma_j$.

(6) The challenger runs $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out}_{\mathsf{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathbf{s}^b)$ and sends $(\{\mathsf{sh}_i\}_{P_i \in B}, \mathsf{out}_{\mathsf{pub}})$ to $\mathcal{A}_3$.

(7) $\mathcal{A}_3$ outputs a bit b'.

(8) The output of the game is defined to be 1 if $b = b'$ and 0 otherwise. We write $\mathrm{MSSS}_{\mathcal{A}_3}^{\mathrm{CSA}}(\lambda) = 1$ if the output is 1 and in this case we say that $\mathcal{A}_3$ succeeded.

A multi-stage secret sharing scheme $\Omega_1 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$, has indistinguishability against chosen secret attacks (or is CSA-secure) in the computational scenario, if for any polynomial-time adversary $\mathcal{A}_3$ there exists a negligible function negl such that

$$\Pr[\mathrm{MSSS}_{\mathcal{A}_3}^{\mathrm{CSA}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

3.2 Formal Model of Computational GMSS

In a GMSS the dealer wants to share l secrets among n participants according to the access structure Γ, such that any authorized set of participants can reconstruct all of the secrets simultaneously in one stage. In the following, we will provide the definition of a CSA-secure computational GMSS and the game-based security definition model.

3.2.1 General Multi-secret Sharing Schemes

A general multi-secret sharing scheme is a tuple of $\Omega_2 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$ such that:

- The setup algorithm Stp takes as input a security parameter 1^λ, the set of participants $\mathcal{P}$ and an access structure Γ and outputs some public and common parameters pms for the scheme; $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, \Gamma)$.
- The distribution algorithm Dist takes as input pms and the global secret $\mathbf{s} = (s_1, \ldots, s_l)$ to be shared, and generates the set of shares $\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}$ and possibly some pubic output $\mathsf{out}_{\mathsf{pub}}$; $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out}_{\mathsf{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathbf{s})$.
- The reconstruction algorithm Rec takes as input $\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}$, and the shares $\{\mathsf{sh}_i\}_{P_i \in A}$ of the participants in some subset $A \subset \mathcal{P}$ and outputs a possible value $\mathbf{s}' = (s_1', \ldots, s_l')$; $\mathbf{s}' := \mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, \{\mathsf{sh}_i\}_{P_i \in A})$.

For correctness, we require that for any subset $A \in \Gamma$ and any $\mathbf{s}$ it holds $\mathbf{s} = \mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, \{\mathsf{sh}_i\}_{P_i \in A})$.

3.2.2 Chosen Secret Attack Indistinguishability Experiment

We now define a game for any general multi secret sharing scheme $\Omega_2 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$, between an adversary $\mathcal{A}_4$ and a challenger.

Game $\mathcal{G}_4$

(1) The challenger chooses a random bit $b \in \{0,1\}$.

(2) $\mathcal{A}_4$ publishes the set of participants $\mathcal{P}$ and access structure Γ.

(3) The challenger runs $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, t)$ and sends pms to $\mathcal{A}_4$.

(4) $\mathcal{A}_4$ broadcasts a subset $B \subset \mathcal{P}$ of corrupted participants such that $B \notin \Gamma$.

(5) $\mathcal{A}_4$ broadcasts two different global secrets $\mathbf{s}^0 = (s_1^0, \ldots, s_l^0) \neq (s_1^1, \ldots, s_l^1) = \mathbf{s}^1$.

(6) The challenger run $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out}_{\mathsf{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathbf{s}^b)$ and sends $(\{\mathsf{sh}_i\}_{P_i \in B}, \mathsf{out}_{\mathsf{pub}})$ to $\mathcal{A}_4$.

(7) $\mathcal{A}_4$ outputs a bit b'.

(8) The output of the game is defined to be 1 if $b = b'$ and 0 otherwise. We write $\mathrm{GMSS}_{\mathcal{A}_4}^{\mathrm{CSA}}(\lambda) = 1$ if the output is 1 and in this case we say that $\mathcal{A}_4$ succeeded.

A general multi-secret sharing scheme $\Omega_2 = (\mathsf{Stp}, \mathsf{Dist}, \mathsf{Rec})$, has indistinguishability against chosen secret attacks (or is CSA-secure) in the computational scenario, if for any polynomial-time adversary $\mathcal{A}_4$ there exists a negligible function negl such that

$$\Pr[\mathrm{GMSS}_{\mathcal{A}_4}^{\mathrm{CSA}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

4 The Computational MSSST2 Scheme

In this section we propose a computational MSSST2, Ω_3 with provable security in the standard model. For simplicity, we consider the case where all the access structures are threshold ones.

4.1 Setup: $\mathsf{Stp}(1^\lambda, \mathcal{P}, t_1, \ldots, t_l)$

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of n participants. A dealer D wants to share l secrets $s_1, \ldots, s_l$ among the participants of $\mathcal{P}$ in such a way that any t_j or more participants can recover the secret s_j, while no $t_j - 1$ participants can obtain any information about the secret s_j and let $1 \leq t_1 \leq t_2 \leq \cdots \leq t_l \leq n$ (because $\Gamma_i \subseteq \Gamma_{i-1}$, for $i = 2, \ldots, l$). D chooses a secure private-key encryption scheme $\Pi_1 = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with key space $\mathcal{K}$, plaintext space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, such that $\mathcal{M}$ contains the space of the secrets to be shared. Let q be a prime number, $q > n$ such that $\mathbb{Z}_q \subset \mathcal{M}$. Each participant P_i is assigned the value i. The public parameters are $\mathsf{pms} = (q, \Pi_1, \mathcal{P}, t_1, \ldots, t_l)$.

4.2 Distribution of the Shares: $\mathsf{Dist}(\mathsf{pms}, \mathsf{s})$

D performs the following steps to share the l secrets $(s_1, \ldots, s_l) \in \mathbb{Z}_q$ among n participants:

(1) Run $k_i \leftarrow \mathsf{Gen}(1^\lambda)$ for $i = 1, \ldots, n$.

(2) The secret share $\mathsf{sh}_i = k_i$ is sent to participant P_i via a secure channel.

(3) Select random polynomials $f_j(x) \in \mathbb{Z}_q[x]$ of degree $t_j - 1$ such that $f_j(0) = s_j$ for $j = 1, \ldots, l$.

(4) Compute the values $c_{i1} = \mathsf{Enc}(k_i, f_1(i))$ and $c_{ij} = \mathsf{Enc}(k_i, f_j(i + s_{j-1}))$, for $1 \leq i \leq n$, $j = 2, \ldots, l$.

(5) The public output of the protocol is $\mathsf{out}_{\mathsf{pub}} = \{c_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq l}$.

4.3 Reconstruction of the Secrets: $\mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, \{\mathsf{sh}_i\}_{P_i \in A})$

The secrets should be reconstructed in the following order: $s_1, s_2, \ldots, s_l$. Now we show that how the participants of an authorized subset $A \subseteq \mathcal{P}$ (i.e. $|A| \geq t_j$) can recover the secret s_j :

(1) If $j = 1$
- Each $P_i \in A$ computes $f_1(i) = \mathsf{Dec}(\mathsf{sh}_i, c_{i1})$.
- Use the pairs $\{(i, f_1(i))\}_{P_i \in A}$ to interpolate the polynomial $f_1(x)$ and recover the secret s_1.

(2) If $j \geq 2$
- Each $P_i \in A$ computes $f_j(i + s_{j-1}) = \mathsf{Dec}(\mathsf{sh}_i, c_{ij})$.
- Use the pairs $\{(i + s_{j-1}, f_j(i + s_{j-1}))\}_{P_i \in A}$ to interpolate the polynomial $f_j(x)$ and recover the secret s_j.

5 Security Analysis of the Computational MSSST2 Scheme

In this section we are going to reduce the computational security of the described computational MSSST2 scheme, Ω_3 to the security of the underlying private-key encryption scheme Π_1 and prove that if Π_1 has indistinguishability against chosen plaintext attacks, then Ω_3 has indistinguishability against chosen secret attacks. As far as we know, this is the first security analysis for MSSST2 in the computational setting. Although we describe and analyze the scheme in the setting of different threshold access structures, it can be easily extended to work with more general access structures.

Theorem 1. *For any adversary $\mathcal{A}_3$ against the chosen secret attacks security of MSSST2, Ω_3 that chooses t^* corrupts participants in a set $\mathcal{P}$ of n participants and chooses global secrets $\mathbf{s}^0 = (s_1^0, \ldots, s_l^0) \neq (s_1^1, \ldots, s_l^1) = \mathbf{s}^1$, there exists a $(\kappa, q_e, q_c)-$adversary $\mathcal{A}_2$ against the chosen plaintext attacks security of private key encryption scheme Π_1 with parameters $\kappa = n - t^*$ and $q_e + q_c = l(n - t^*)$ such that*

$$\Pr[\mathrm{MSSS}_{\mathcal{A}_3}^{\mathrm{CSA}}(\lambda) = 1] = \Pr[\mathrm{PrivK}_{\mathcal{A}_2}^{\mathrm{CPA}}(\lambda) = 1]$$

Proof. The proof is by reduction. Let $\mathcal{A}_3$ be an adversary against the computational security of the described threshold MSSST2 scheme Ω_3. We are going to construct an adversary $\mathcal{A}_2$ against the CPA security of private-key encryption scheme Π_1, which will use $\mathcal{A}_3$ as a sub-routine as follow:

(1) The challenger of the game $\mathcal{G}_2$ starts this game by choosing a random bit $\beta \in \{0, 1\}$.
- $\mathcal{A}_3$ starts the game $\mathcal{G}_3$ by choosing the set of participants $\mathcal{P}$ and threshold values $t_1, \ldots, t_l$ s.t. $1 \leq t_1 \leq t_2 \leq \cdots \leq t_l \leq n$.
- $\mathcal{A}_2$ acts as the challenger of the security game $\mathcal{G}_3$ and has to simulate running of the $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, t_1, \ldots, t_l)$. To do this, $\mathcal{A}_2$ chooses a prime number $q > n$, such that $\mathbb{Z}_q \subset \mathcal{M}$, and sends $\mathsf{pms} = (q, \Pi, \mathcal{P}, t_1, \ldots, t_l)$ to $\mathcal{A}_3$.
- $\mathcal{A}_3$ broadcasts a subset $B \subset \mathcal{P}$ of corrupted participants such that $|B| = t^*$. Without loss of generality, we assume that $B = \{P_1, \ldots, P_{t^*}\}$.
- $\mathcal{A}_3$ broadcasts two different global secrets $\mathbf{s}^0 \neq \mathbf{s}^1$ with the following restriction: $s_j^0 = s_j^1$ for j s.t. $t^* \geq t_j$. Let $\mathbb{J}^* = \{j \in \{1, \ldots, l\}$ *s.t.* $s_j^0 = s_j^1\}$. Note that if $t^* \geq t_j$, then $j \in \mathbb{J}^*$.
- $\mathcal{A}_2$ runs $k_i \leftarrow \mathsf{Gen}(1^\lambda)$ for $P_i \in B$

(2) $\mathcal{A}_2$ defines the number $\kappa = n - t^*$ of keys in the game $\mathcal{G}_2$.

(3) The challenger of game $\mathcal{G}_2$, runs κ times $\mathsf{Gen}(1^\lambda)$, to produce κ secret keys $k_{t^*+1}, \ldots, k_n$ for the non-corrupted participants $P_i \notin B$.

- For each $j \in \mathbb{J}^*$, $\mathcal{A}_2$ chooses random polynomials $f_j(x) \in \mathbb{Z}_q[x]$ of degree $t_j - 1$ s.t. $f_j(0) = s_j^0 = s_j^1$.
- For each $P_i \in B$ and each $j \in \mathbb{J}^*$, $\mathcal{A}_2$ computes the value c_{ij} in two different ways: If $j = 1$, then $c_{i1} = \mathsf{Enc}\big(k_i, f_1(i)\big)$, and if $j > 1$, then $c_{ij} = \mathsf{Enc}\big(k_i, f_j(i + s_{j-1})\big)$.

(4) For each $P_i \notin B$ and each $j \in \mathbb{J}^*$, $\mathcal{A}_2$ sends the following encryption query: If $j = 1$, then $\big(i, f_1(i)\big)$, and if $j > 1$, then $\big(i, f_j(i + s_{j-1})\big)$ to its encryption oracle. *This means that $\mathcal{A}_2$ has made $q_e = |\mathbb{J}^*|(n - t^*)$ encryption queries.*

(5) For each $P_i \notin B$ and each $j \in \mathbb{J}^*$, $\mathcal{A}_2$ receives the following value c_{ij}: If $j = 1$, then $c_{i1} = \mathsf{Enc}\big(k_i, f_1(i)\big)$, and if $j > 1$, then $c_{ij} = \mathsf{Enc}\big(k_i, f_j(i + s_{j-1})\big)$.

- For each $j \notin \mathbb{J}^*$, $\mathcal{A}_2$ chooses random pairs of polynomials $f_j^0(x), f_j^1(x) \in \mathbb{Z}_q[x]$ of degree $t_j - 1$ in two different ways: If $j = 1$, then $f_1^0(0) = s_1^0, f_1^1(0) = s_1^1$, and $f_1^0(i) = f_1^1(i)$ for each $P_i \in B$, and if $j > 1$, then $f_j^0(0) = s_j^0, f_j^1(0) = s_j^1$, and $f_j^0(i + s_{j-1}) = f_j^1(i + s_{j-1})$ for each $P_i \in B$.
- For each $P_i \in B$ and each $j \notin \mathbb{J}^*$, $\mathcal{A}_2$ computes the value c_{ij} in two different ways: If $j = 1$, then $c_{i1} = \mathsf{Enc}\big(k_i, f_1^0(i)\big)$, an if $j > 1$, then $c_{ij} = \mathsf{Enc}\big(k_i, f_j^0(i + s_{j-1})\big)$.

(6) For each $P_i \notin B$ and each $j \notin \mathbb{J}^*$, $\mathcal{A}_2$ sends the following challenger query: If $j = 1$, then $\big(i, f_1^0(i), f_1^1(i)\big)$, and if $j > 1$, then $\big(i, f_j^0(i + s_{j-1}), f_j^1(i + s_{j-1})\big)$ to its challenger (game $\mathcal{G}_2$). *So, the number of challenge queries made by $\mathcal{A}_2$ is $q_c = (l - |\mathbb{J}^*|)(n - t^*)$.*

(7) For $j \notin \mathbb{J}^*, P_i \notin B$ the challenger of the game $\mathcal{G}_2$, computes c_{ij} in two different ways: If $j = 1$, then runs $c_{i1} \leftarrow \mathsf{Enc}\big(k_i, f_1^\beta(i)\big)$, and if $j > 1$, then $c_{ij} \leftarrow \mathsf{Enc}\big(k_i, f_j^\beta(i + s_{j-1})\big)$. Challenger sends back c_{ij} to $\mathcal{A}_2$.

- $\mathcal{A}_2$ publishes the public output $\mathsf{out_{pub}} = \{c_{ij}\}_{1 \le i \le n, 1 \le j \le l}$ and sends the secret shares $\{\mathsf{sh}_i\}_{P_i \in B}$ of the corrupted participants, defined as $\mathsf{sh}_i = k_i$. In this way, $\mathcal{A}_2$ is perfectly simulating an execution of the distribution protocol $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out_{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathbf{s}^b)$, where $b = \beta$ and $\mathbf{s}^b = (s_1^b, \ldots, s_l^b)$.
- $\mathcal{A}_3$ outputs a bit $b' \in \{0, 1\}$.

(8) $\mathcal{A}_2$ outputs the same bit $\beta' = b'$.

Thus, we have

$$\begin{aligned}\Pr[\mathrm{PrivK}^{\mathrm{CPA}}_{\mathcal{A}_2}(\lambda) = 1] &= \Pr[\beta' = \beta] \\ &= \Pr[b' = b] \\ &= \Pr[\mathrm{MSSS}^{\mathrm{CSA}}_{\mathcal{A}_3}(\lambda) = 1].\end{aligned}$$

The second equality above is due to $\beta' = b'$ and $\beta = b$. This completes the proof. □

5.1 Multi-stage Feature

Obviously, our scheme is based on the Lagrange interpolation polynomial. At least t_j participants must provide their shares and reconstruct the secret s_j in $j-$th stage. If they do not have the previous secret s_{j-1} first, they cannot obtain the secret s_j. For this reason, they must reconstruct the secrets in the special order: $s_1, s_2, \ldots, s_l$.

6 The Computational GMSS Scheme

In this section we provide a computational GMSS, Ω_4 and show that it has provable security in the standard model. To the best of our knowledge, this is the first security analysis for GMSS in the computational setting. For simplicity, we consider the case where all the access structures are threshold ones. Ω_4 enjoys the same level of security of Ω_3 when an M.Eav-secure encryption is used in its construction. We now give the details.

6.1 Setup: $\mathsf{Stp}(1^\lambda, \mathcal{P}, t)$

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of n participants. A dealer D wants to share l secrets among the participants of $\mathcal{P}$ in such a way that any t or more participants can recover the secrets, while no $t - 1$ participants can obtain any information about the secrets. D chooses a secure private-key encryption scheme $\Pi_2 = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with key space $\mathcal{K}$, plaintext space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, such that $\mathcal{M}$ contains the space of the secrets to be shared. Let q be a prime number and $q > n$ such that $\mathcal{K} \subset \mathbb{Z}_q$. Each participant P_i is assigned the value i. The public parameters are $\mathsf{pms} = (q, \Pi_2, \mathcal{P}, t)$.

6.2 Distribution of the Shares: $\mathsf{Dist}(\mathsf{pms}, \mathbf{s})$

D performs the following steps to share the l secrets $(s_1, \ldots, s_l) \in \mathcal{M}^l$ among n participants:

(1) Run $k \leftarrow \mathsf{Gen}(1^\lambda)$.

(2) Compute $c_j \leftarrow \mathsf{Enc}(k, s_j)$, for $j = 1, \ldots, l$.

(3) Select a random polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree $t - 1$ such that $f(0) = k$.

(4) Compute the values $k_i = f(i)$, for $1 \le i \le n$.

(5) The secret share $\mathsf{sh}_i = k_i$ is sent to player P_i via a secure channel, whereas the public output of

the protocol is $\mathsf{out}_{\mathsf{pub}} = \{c_1, \ldots, c_l\}$.

6.3 Reconstruction of the Secrets: $\mathsf{Rec}(\mathsf{pms}, \mathsf{out}_{\mathsf{pub}}, \{\mathsf{sh}_i\}_{P_i \in A})$

Now we show that how the participants of an authorized subset $A \subseteq \mathcal{P}$ (i.e. $|A| \geq t$) can recover the secrets simultaneously in one stage:

(1) Use the pairs $\{(i, \mathsf{sh}_i)\}_{P_i \in A}$ to interpolate the polynomial $f(x)$ and recover the value k.

(2) Take the values $\{c_i\}_{1 \leq i \leq l}$ from $\mathsf{out}_{\mathsf{pub}}$ and compute the secrets $s_i = \mathsf{Dec}(k, c_i)$ for $1 \leq i \leq l$.

7 Security Analysis of the Computational GMSS Scheme

In this section we are going to reduce the computational security of the described threshold scheme Ω_4 to the security of the underlying private-key encryption scheme Π_2 and prove that if Π_2 has indistinguishable multiple encryptions in the presence of an eavesdropper, then Ω_4 has indistinguishability against chosen secret attacks. Although we describe and analyze the scheme for the case of threshold access structure, it can be easily extended to more general access structure.

Theorem 2. *For any adversary $\mathcal{A}_4$ against the described threshold computational GMSS, Ω_4, that chooses corrupts participants in a set $\mathcal{P}$ and chooses global secrets $\mathbf{s}^0 = (s_1^0, \ldots, s_l^0) \neq (s_1^1, \ldots, s_l^1) = \mathbf{s}^1$, there exists a q_c-adversary $\mathcal{A}_1$ against the eavesdropper attacks security of private-key encryption scheme Π_2, where $q_c = l$ and*

$$\Pr[\mathrm{GMSS}_{\mathcal{A}_4}^{\mathrm{CSA}}(\lambda) = 1] = \Pr[\mathrm{PrivK}_{\mathcal{A}_1}^{\mathrm{M.Eav}}(\lambda) = 1]$$

Proof. Again, the proof is by reduction. Using a similar way we can show that if $\mathcal{A}_4$ is an adversary against the computational security of the Ω_4 then it is possible to construct an adversary $\mathcal{A}_1$ against the multiple eavesdropper attacks security of private-key encryption scheme Π_2 which uses $\mathcal{A}_4$ as a sub-routine.

(1) The challenger of the game $\mathcal{G}_1$, chooses a random bit $\beta \in \{0, 1\}$ and runs $\mathsf{Gen}(1^\lambda)$, to produce a secret key k.

- $\mathcal{A}_4$ starts the game $\mathcal{G}_4$ by choosing the set of participants $\mathcal{P}$ and threshold value t.
- $\mathcal{A}_1$ acts as the challenger of the security game $\mathcal{G}_2$ and has to simulate running of the $\mathsf{pms} \leftarrow \mathsf{Stp}(1^\lambda, \mathcal{P}, t)$. To do this, $\mathcal{A}_1$ chooses a prime number $q > n$ such that the key space $\mathcal{K}$ of the target private-key encryption scheme Π_2 satisfies $\mathcal{K} \subset \mathbb{Z}_q$, and sends $\mathsf{pms} = (q, \Pi_2, \mathcal{P}, t)$. to $\mathcal{A}_4$.
- $\mathcal{A}_4$ broadcasts a subset $B \subset \mathcal{P}$ of corrupted participants such that $|B| < t$.
- $\mathcal{A}_4$ broadcasts two different global secrets $\mathbf{s}^0 \neq \mathbf{s}^1$ with the following restriction: $|s_j^0| = |s_j^1|$ for $j = 1, \ldots, l$.
- For each $P_i \in B$, $\mathcal{A}_1$ chooses at random $\mathsf{sh}_i = k_i \in \mathbb{Z}_q$. The values $\{k_i\}_{P_i \in B}$ are perfectly possible shares of the unknown secret key k.

(2) $\mathcal{A}_1$ sends tuples $\mathbf{s}^0 = (s_1^0, \ldots, s_l^0)$ and $\mathbf{s}^1 = (s_1^1, \ldots, s_l^1)$ to its challenger (game $\mathcal{G}_1$).

(3) The challenger runs $c_i \leftarrow \mathsf{Enc}(k, s_i^\beta)$ and sends back $(c_1, \ldots, c_l)$ to $\mathcal{A}_1$.

- $\mathcal{A}_1$ publishes the public output $\mathsf{out}_{\mathsf{pub}} = \{c_1, \ldots, c_l\}$ and sends $\{\mathsf{sh}_i\}_{P_i \in B}$ to $\mathcal{A}_4$. In this way, $\mathcal{A}_1$ is perfectly simulating an execution of the distribution protocol $(\{\mathsf{sh}_i\}_{P_i \in \mathcal{P}}, \mathsf{out}_{\mathsf{pub}}) \leftarrow \mathsf{Dist}(\mathsf{pms}, \mathbf{s}^b)$, where $b = \beta$ and $\mathbf{s}^b = (s_1^b, \ldots, s_l^b)$.
- $\mathcal{A}_4$ outputs a bit $b' \in \{0, 1\}$.

(4) $\mathcal{A}_1$ outputs the same bit $\beta' = b'$.

Thus, we have

$$\begin{aligned}\Pr[\mathrm{PrivK}_{\mathcal{A}_1}^{\mathrm{M.Eav}}(\lambda) = 1] &= \Pr[\beta' = \beta] \\ &= \Pr[b' = b] \\ &= \Pr[\mathrm{GMSS}_{\mathcal{A}_4}^{\mathrm{CSA}}(\lambda) = 1].\end{aligned}$$

The second equality above is due to $\beta' = b'$ and $\beta = b$. This completes the proof. □

7.1 General Feature

Since the secrets can only be recovered by $s_i = \mathsf{Dec}(k, c_i)$ for $1 \leq i \leq l$, at least t participants can obtain the private key k and reconstruct all of the secrets simultaneously.

8 Comparative Results

We have compared the proposed multi-secret sharing schemes, Ω_3, Ω_4 with Herranz *et al.*'s scheme [9] in Table 1. It should be noted that Ω_3 and Herranz *et al.*'s schemes belong to multi-stage secret sharing schemes, while Ω_4 is a GMSS. Moreover, the secrets should be reconstructed according to a predefined order in Ω_3, while the secret reconstruction of Herranz *et al.*'s scheme can be executed in any order. We do not consider other multi-secret sharing schemes in Table 1, because they lack a formal security analysis. Also, in Table 2 we compare the proposed schemes with MSSST2 and GMSSs proposed in [5, 7, 10, 15, 17, 20]. According to the specific example with $l = 32, n = 2^{10} = 1024, q = 2^{11} = 2048$ and results in [9], we obtain the length of sh_i's in Table 2.

Table 1. Basic comparison between the Herranz *et al.* and our schemes

Property	Herranz *et al.* [9]	Ω_3	Ω_4
Space of secret $\mathbf{s}$	$\mathcal{M}^l$	$\mathcal{M}^l$	$\mathcal{M}^l$
Length of $\mathsf{out}_{\mathsf{pub}}$	$nl\|\mathcal{C}\|$	$nl\|\mathcal{C}\|$	$l\|\mathcal{C}\|$
Length of sh_i	$\|\mathcal{K}\|$	$\|\mathcal{K}\|$	$\|\mathcal{K}\|$
Security level of private-key encryption Π	CPA	CPA	M.Eav
Security level of MSS	IND-CSA	IND-CSA	IND-CSA
Category of MSS	MSSST1	MSSST2	GMSS
Security model	SM	SM	SM
Recover multi-secrets parallelly	No	No	Yes
Participants can recover only one secret in every stage	Yes	Yes	No
The secret reconstruction must be executed in a predefined order	No	Yes	No
Size of each share is shorter than size of secret	Yes	Yes	Yes
Has indistinguishability against chosen secret attacks	Yes	Yes	Yes

Table 2. Basic comparison between the MSSST2 & GMSS

Property	Chang [5]	Li [15]	Ω_3	Yang [20]	Hu [10]	Hadian [7]	Mashhadi [17]	Ω_4
Length of sh_i in 2048-bit finite fields	2048	2048	149	2048	680	2048	680	149
Number of public values	ln	$l(n-t)$	ln	$n+l$	$2n+l$	$2n+l$	$2n+l$	l
Category of MSS	MSSST2	MSSST2	MSSST2	GMSS	GMSS	GMSS	GMSS	GMSS
Security model	-	-	SM	-	-	-	-	SM
Recover multi-secrets parallelly	No	No	No	Yes	Yes	Yes	Yes	Yes
Participants can recover only one secret in every stage	Yes	Yes	Yes	No	No	No	No	No
The secret reconstruction must be executed in a predefined order	Yes	Yes	Yes	No	No	No	No	No
Size of each share is shorter than size of secret	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

8.1 General Access Structures

We, similar to Herranz, have described and analyzed the two new schemes in the setting of threshold access structures. These schemes can be easily extended to work with more general access structures. We just have to replace the use of Shamir's threshold secret sharing scheme with the use of some other secret sharing scheme which supports more general access structures, such as monotone span programs [11].

8.2 Applications

MSSST1s such as Herranz *et al.*'s scheme are very useful for situations where different running of a secret task (like signature or decryption) may have different levels of importance or security [9]. Beside, in the real world applications, MSSST2s such as Ω_3 are very practical. For example, there may be a security system of bank's confidential database where one must pass through l checkpoints before the database can be accessed. To distribute the power of a single authority and the security policy, the checkpoints should be opened and passed in sequence by at least t participants together. If the checkpoints (secrets) do not follow the proper order, it will harm the security of the system [5].

Moreover, GMSSs such as Ω_4 are useful in several other kinds of applications [20]: Sometime it is required that several secrets be protected with the same amount of data usually needed to protect one secret, or sometimes people need to partition one large secret into l pieces with each piece protected by a smaller amount of data than is needed to protect the entire secret.

References

[1] M. Bellare, A. Boldyreva, S. Micali, *Public-key encryption in a multi-user setting: Security proofs and improvements*, in: Proceeding of Eurocrypt'00, in: LNCS, 1807, Springer-Verlag, 2000, pp. 259-274.

[2] M. Bellare, A. Desai, E. Jokipii, P. Rogaway, *A concrete security treatment of symmetric encryp-*

tion, in: FOCS'97, IEEE Society Press, 1997, pp. 394-403.

[3] M. Bellare, P. Rogaway, *Introduction to modern cryptography*, Notes for the course CSE207 of the University of California, San Diego, available at http://cseweb.ucsd.edu/classes/sp99/cse207/index.html, visited at November 2012.

[4] C. Cachin, *On-line secret sharing,* in Proceedings of IMA Conference'95, in: LNCS, 1025, Springer-Verlag, 1995, pp. 190-198.

[5] T.Y. Chang, M. S. Hwang, W. P.Yang, *A new multi-stage secret sharing scheme using one-way function*, ACM SIGOPS Operating Systems, 39, 2005, pp. 48-55.

[6] H.-Y. Chien, J.-K. Jan, Y.-M. Tseng, *A practical* (t, n) *multi-secret sharing scheme*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer 83-A, 12, 2000, pp. 2762-2765.

[7] M. H. Dehkordi, S. Mashhadi, *New efficient and practical verifiable multi-secret sharing schemes*, Information Sciences 178, 9, 2008, pp. 2262-2274.

[8] J. He, E. Dawson, *Multistage secret sharing based on one-way function*, Electronics Letters 30, 19, 1994, pp. 1591-1592.

[9] J. Herranz, A. Ruiz, G. Sáez, *Sharing many secrets with computational provable security*, Information Processing Letters 113, 2013, pp. 572-579.

[10] C. Hu, X. Liao, X. Cheng, *Verifiable multi-secret sharing based on LFSR sequences*, Theoret. Commun. Sci. 445, 2012, pp. 52-62.

[11] M. Karchmer, A. Wigderson, *On Span Programs*, in: Proceeding of SCTC'93,IEEE Computer Society Press, 1993, pp. 102-111.

[12] J. Katz, Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall/CRC, New York, 2008.

[13] J. Katz, M. Yung, *Characterization of security notions for probabilistic private-key encryption*, Journal of Cryptology, 19, 2006, pp. 67-95.

[14] H. Krawczyk, *Secret sharing made short, in: Proceedings of Crypto'93*, in LNCS, 773, Springer-Verlag, 1993, pp. 136-146.

[15] H. X. Li, C. T. Cheng, L. J. Pang, *An improved Multi-stage* (t, n)*-threshold secret sharing scheme*, LNCS 3739, 2005, pp. 267-274.

[16] H. Y. Lin, Y. S. Yeh, *Dynamic multi-secret sharing scheme*, Int. J. Contemp. Math. Sciences, 3, 2008, pp. 37-42.

[17] S. Mashhadi, M. Hadian Dehkordi, *Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem*, Information Sciences 294, 2015, pp. 31-40.

[18] B. Masucci, *Sharing multiple secret: Models, schemes and analysis*, Designs, Codes and Cryptography, 39, 2006, pp. 89-111.

[19] A. Shamir, *How to share a secret*, Communications of the ACM. 22, 1979, pp. 612-613.

[20] C.-C. Yang, T.-Y. Chang, M.-S. Hwang, *A* (t, n) *multi-secret sharing scheme*, Applied Mathematics and Computation 151, 2004, pp. 483-490.

Samaneh Mashhadi was born in Tafresh, Iran, on March 27, 1982. She received the B.Sc. and M.Sc. degrees with honors in Mathematics from Iran University of Science and Technology (IUST), and Amirkabir University of Technology (AUT) in 2003 and 2005, respectively. She received her Ph.D. with honors in Mathematics (Cryptography) in 2008 from IUST. She is currently an assistant professor in Department of Mathematics of IUST. Her research interests include analysis, design, and application of digital signatures, secret sharing schemes, and security protocols.