Scientia Iranica D (2017) 24(3), 1448–1457

SCIENTIA IRANICA

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.scientiairanica.com

A lattice-based changeable threshold multi-secret sharing scheme and its application to threshold cryptography

H. Pilaram[a] and T. Eghlidos[b,*]

a. *School of Electrical Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-8639, Iran.*
b. *Electronics Research Institute, Sharif University of Technology, Tehran, P.O. Box 11155-8639, Iran.*

Received 6 March 2016; received in revised form 17 August 2016; accepted 29 October 2016

KEYWORDS

Threshold multi-stage secret sharing; Changeable threshold secret sharing; Threshold decryption; Lattice-based cryptography.

Abstract. In this paper, we propose a threshold increasing algorithm for a (t, n) lattice-based Threshold Multi-Stage Secret Sharing (TMSSS) scheme. To realize the changeability feature, we use the zero addition protocol to construct a new (t', n) TMSSS scheme. Therefore, the new scheme enjoys the significant feature of threshold changeability along with the inherited features of being multi-stage, multi-use, and verifiable derived from our previously proposed lattice-based TMSSS scheme. Furthermore, we use the improved TMSSS scheme to propose a threshold decryption algorithm for the Learning With Error (LWE) based public key encryption scheme based on the study of Lindner and Peikert. For threshold decryption, each authorized subset of participants decrypts the ciphertext partially and sends the result to the combiner. The combiner can decrypt the ciphertext using the partial decryptions. The security of both schemes is based on hardness of lattice problems, i.e. LWE and Inhomogeneous Small Integer Solution (ISIS) problems, which are believed to resist against the quantum algorithms. The proposed schemes are efficient, especially on the participants' side, making them suitable for the applications in which the participants have limited processing capacities.

1. Introduction

Secret sharing is a cryptographic primitive with many applications such as key management in sensor networks [1], electronic cash [2], electronic voting [3], and cloud computing [4]. A secret sharing scheme splits a secret among a set of parties, called participants, in such a way that some authorized subsets of the participants can reconstruct the secret using their assigned values, called shares, by a trusted third party named dealer. A (t, n) Threshold Secret-Sharing Scheme (TSSS) is a special case of secret sharing schemes, in which at least t participants are required to recover the secret. The first of such a scheme was introduced by Blakley and Shamir, independently, in 1979 [5,6]. Since then, some new features have been added to the secret sharing schemes such as verifiability of the shares [7,8], resistance of the scheme in the presence of a number of cheaters [9,10], and dynamic change of the threshold and/or the number of participants [11,12].

To share more than one secret, multi-secret sharing schemes have been introduced, which is a generalization of the secret sharing schemes [13]. In these schemes, each participant is given one share to recover all the secrets, the size of which is the same as the size of the secrets. These schemes only provide computational security [14]. Pang et al. [15] proposed a multi-secret sharing scheme for general access structure

*. *Corresponding author. Tel.: +98 21 66164960*
E-mail addresses: h pilaram@ee.sharif.edu (H. Pilaram); teghlidos@sharif.edu (T. Eghlidos)

in 2006. In this scheme, when an authorized subset of participants pulls their shares together, all of the secrets are revealed at the same time. In 1994, He and Dawson [16] proposed a (t, n) TMSSS scheme. A multi-secret sharing scheme is called multi-stage if the secrets are not revealed at the same time, i.e. in recovering a number of secrets, the recovered secrets do not leak any information about the unrecovered secrets. In 2007, Geng et al. [17] showed that the He-Dawson's scheme is actually one-time-use and vulnerable to collusion attacks. They proposed a multi-use threshold secret sharing scheme using a one-way hash function. The term "multi-use" in this paper implies that the same shares are used by some technical measures when a new set of secrets is to be shared. For this purpose, the following two security requirements are to be realized:

1. While recovering the secret(s), the participants must not reveal the original shares;

2. The secrecy of the other unrecovered secrets should be computationally independent from the recovered secrets.

For constructing a TMSSS scheme, the participants should send the pseudo-secret shares instead of the original ones to the combiner, in which the pseudo-secret shares depend on the original shares and the desired secret which is to be recovered. All the existing TMSSS schemes are based on one-way (hash) functions [14,18,19], two-variable one-way functions [20,21], and assumptions such as difficulty of solving discrete logarithm problem [22], which can now be threatened by quantum algorithms.

In a TSSS, the importance of the secret as well as the mutual trust between the participants or the organizational structure of the participants may vary. Hence, it might be required that the TSSS be changed in such a way that the larger number of participants are needed to recover the secret. Therefore, the threshold should be increased. A Changeable TSSS (CTSSS) is a TSSS in which the shares generated initially by the dealer should be changed in such a way that the secret can be recovered with a larger threshold t' (i.e., $t < t'$). In 1999, Martin et al. proposed the first CTSSS [11]. Later on, different CTSSSs have been proposed. Such schemes are classified into three types: The schemes based on a linear polynomial [23,24], the hyperplane geometrical based schemes [11], and the schemes based on the Chinese Reminder Theorem (CRT) [25,26]. In 2004, Steinfeld et al. [23] proposed a lattice-based CTSSS to increase the threshold in the standard Shamir's secret sharing scheme. Their scheme is dealer-free which does not need any secure channel. However, their scheme uses a lattice-based method to recover the secret instead of the conventional polynomial interpolation used in the Shamir's TSSS.

The security of the currently used public key cryptosystems based on "integer factorization" and "discrete logarithm" has been threatened since invention of Shor's quantum algorithm in 1994 [27]. Ever since Ajtai's introduction of the lattice-based one-way functions [28], the field of lattice-based cryptography plays a significant role in the world of post-quantum cryptography. Ajtai proposed a family of one-way functions whose security is based on the worst-case hardness of the lattice problems such as Shortest Vector Problem (SVP) known to be NP-hard [29]. Lattice-based cryptosystems enjoy provable security based on the worst-case hardness of lattice problems. Furthermore, they only need linear computations on relatively small integers.

In 2005, Regev proposed a public key cryptosystem based on the LWE problem [30]. Regev and Peikert independently showed that, for certain parameters, LWE is as hard as classical lattice problems, such as the Shortest Independent Vector Problem (SIVP), in the worst case. It follows that LWE-based schemes are provably secure assuming the worst-case hardness of classical lattice problems. The LWE problem has been the source of great progress in the lattice-based cryptography. In 2010, Lindner and Peikert proposed an LWE-based cryptosystem enjoying the advantages of having substantially smaller key and ciphertext sizes than those of the more well-known cryptosystems proposed in the literature [31].

In recent years, some lattice-based TSSSs have been proposed. Georgescu [32] proposed an (n, n) TSSS based on the hardness of the LWE problem. In 2012, Bansarkhani et al. [33] proposed a verifiable (n, n) TSSS using linear lattice-based hash functions to enable each participant to verify their share as well as the recovered secret. The security of this scheme relies on the hardness of n^c-approximate SVP. Amini et al. [34] and Asaad et al. [35] proposed the first (t, n) TSSSs with asymptotic security in 2014. Later, we have proposed an efficient lattice-based verifiable TMSSS using Ajtai's one-way function in 2015 [36].

In a threshold cryptosystem, the private key is shared among the participants in which at least a certain number of them are required to decrypt or sign the message. Bendlin et al. proposed the first lattice-based threshold cryptographic scheme in threshold decryption of one-bit message [37] based on Regev's LWE public key encryption scheme. Frederiksen extended this scheme for multi-bit messages [38]. Singh et al. [39] proposed an efficient lattice-based threshold public key encryption scheme based on Lindner's work [31]. In 2013, Bendlin et al. [40] proposed a threshold signature and an identity-based encryption scheme. The above-mentioned schemes have used Shamir's TSSS to share the private key as an array. Hence, each entry of the

private key is shared independently in these schemes. However, this method seems to be inefficient.

Contributions. In this paper, the authors' contribution is twofold. First, we propose a threshold increasing algorithm for our previously-introduced lattice-based TMSSS scheme [36]. It should be noted that the already supported features, such as being multi-stage, multi-use and verifiable remain unchanged. For realization of the changeability feature, we share the zero secret with the new threshold and combine the new parameter arrays with the previous ones. Second, using the improved TMSSS scheme, we introduce a threshold decryption algorithm for the LWE-based public key encryption scheme based on Lindner and Peikert's. Here, we consider the column vectors of the private key matrix as the secrets and share them using the proposed lattice-based TMSSS scheme. For threshold decryption, the participants do partial decryption on the ciphertext using their shares and send the results to the combiner. Using additive homomorphic property of the TSSS and linear property of the decryption, the combiner can decrypt the ciphertext using the partial decryptions.

The security of the proposed schemes is based on the hardness of lattice problems which are believed to resist against the quantum algorithms [41]. The proposed threshold public key decryption algorithm inherits the desired features from the improved TMSSS scheme. In the context of threshold public key cryptography, multi-stage feature implies that the participants can decrypt each bit of the message in different stages, where the other bits remain undisclosed. Moreover, both schemes are efficient, especially on the participants' side, because simple matrix operations are used in the secret sharing and threshold decryption protocols. Hence, they are suitable for the mobile applications such as smart cards and sensor networks, in which low processing capability is a dominant factor.

This paper is organized as follows: Section 2 provides a brief review of lattices, our previous multi-stage secret sharing scheme, and the LWE-based public key encryption scheme proposed by Lindner and Peikert. Section 3 presents the proposed lattice-based changeable TMSSS scheme including the algorithm. Section 4 provides the proposed threshold decryption algorithm for Lindner and Peikert's scheme. Sections 5 and 6 discuss the security and efficiency of the proposed schemes, respectively. Finally, a brief conclusion draws all the points together.

2. Preliminaries

In this section, we introduce some basic concepts of lattice, our previous multi-stage secret sharing scheme, and the LWE-based public key encryption scheme proposed by Lindner and Peikert [31].

2.1. Notations

In this paper, we assume column vectors for our case. Lowercase and uppercase letters denote vectors and matrices, respectively. Matrix I_n refers to $n \times n$ identity matrix, and matrix $\mathbf{0}_{m\times n}$ represents zero matrix of size $m \times n$. The transpose of a rectangular matrix is denoted by $(\cdot)^T$. Also, $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{Z}_q$ denote the sets of reals, integers, and the finite field modulo q, respectively. If $\mathbb{S}$ is a set of numbers, $\mathbb{S}^n$ denotes the set of vectors of size n, and $\mathbb{S}^{m\times n}$ denotes the set of $m \times n$ matrices, whose entries are chosen from $\mathbb{S}$. The operator $\lfloor\cdot\rceil$ denotes rounding operation to the nearest integer. The operator $\|\cdot\|$ denotes an arbitrary norm. The most important class of norms is ℓ_p norms, defined for any $p \geq 1$ and a vector $x \in \mathbb{R}^n$ as $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$. The standard big-$O$ and little-o notations are used to classify the growth of functions. Function $\mathrm{negl}(n)$ denotes a negligible function which is defined as $f(n) = o(n^{-c})$ for every fixed constant c.

2.2. Lattices

Here, a lattice is a regular array of points in m-dimensional real vector space.

Definition 1. *[29] Let $b_1, b_2, \cdots, b_n$ be n linearly independent vectors in vector space $\mathbb{R}^m$. $L(b_1, \cdots, b_n)$ is defined to be the set of all integer linear combinations of $b_1, b_2, \cdots, b_n$ as follows:*

$$\Lambda = L(b_1, \cdots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}. \quad (1)$$

The set of vectors $\{b_1, \cdots, b_n\}$ is called a basis for lattice Λ, and n is called the rank of the lattice.

Lattice-based cryptosystems are based on the hardness of lattice problems, SVP and Closest Vector Problem (CVP) are the most popular ones among them [29]. In the lattice-based cryptography, we usually use the approximate version of these problems, denoted by approximation factor, γ. For example, in γ-approximate SVP, we want to find a vector in the lattice whose length is within factor γ of the shortest vector; in γ-approximate CVP, we seek a vector in the lattice whose distance from the target vector is at most γ times that of the closest vector.

Definition 2. *[41] A q-ary lattice is lattice Λ satisfying $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$ for some (possibly prime) integer q.*

For instance, given integer matrix $A \in \mathbb{Z}_q^{n\times m}$ and modulus q, the set of vectors $x \in \mathbb{Z}^m$ satisfying equation $Ax = 0 \mod q$ forms a lattice of dimension m, which is closed under congruence modulo q. This lattice is denoted by $\Lambda_q^{\perp}(A)$.

In [28], Ajtai introduced one-way function $f_A(x) = Ax \bmod q$, where $A \in \mathbb{Z}_q^{n\times m}$ and $x \in \{0,1\}^m$. To invert this function, the following problem is concluded:

Parameters: n, m, $q \in \mathbb{N}$, such that $m > n\log q$, and $q = O(n^c)$ for some constant c;

Input: A uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$ and vector $y = Ax$ for some random vector $x \in \{0,1\}^m$;

Output: A vector $x \in \{0,1\}^m$, such that $Ax = y \bmod q$.

Ajtai proved that solving this problem with non-negligible probability leads to an algorithm which solves any instance of n^c-approximate SVP and is not vulnerable to quantum algorithms in polynomial time.

This problem is a special case of the ISIS problem, in which condition $x \in \{0,1\}^m$ is replaced by $\|x\| \leq \beta$ for real parameter β. Solving ISIS is equivalent to decoding arbitrary integer target point, $t \in \mathbb{Z}^m$, within distance β on q-ary lattice $\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m | Ax = 0 \bmod q\}$, where the syndrome of the target point is $u = At \bmod q$ [42].

2.3. Lindner and Peikert's LWE-based public key encryption scheme

Here, we explain the LWE-based public key encryption scheme proposed by Lindner and Peikert [31]. The scheme uses uniformly random public matrix, $Q \in \mathbb{Z}_q^{n_1\times n_2}$. The cryptosystem uses the following algorithms for the key generation, encryption, and decryption, respectively:

- $Gen(Q, 1^l)$: Choose $R_1 \leftarrow D_{\mathbb{Z},s_k}^{n_1\times l}$ and $R_2 \leftarrow D_{\mathbb{Z},s_k}^{n_2\times l}$ and let $P = R_1 - QR_2 \in \mathbb{Z}_q^{n_1\times l}$. $D_{\mathbb{Z},s_k}$ is the discrete Gaussian distribution on integer numbers with zero mean and standard deviation of s_k. l is the number of bits in message m. Parameters $\{P, Q\}$ and R_2 represent the public and private keys, respectively. The relation between the public and private keys can be written as:

$$\begin{bmatrix} Q & P \end{bmatrix} \begin{bmatrix} R_2 \\ I \end{bmatrix} = R_1 \mod q. \tag{2}$$

- $Enc(Q, P, m \in \{0,1\}^l)$: Choose $e = (e_1, e_2, e_3) \in \mathbb{Z}^{n_1} \times \mathbb{Z}^{n_2} \times \mathbb{Z}^l$ whose entries are chosen from $D_{\mathbb{Z},s_e}$. The ciphertext can be obtained by:

$$\begin{aligned} c^T &= \begin{bmatrix} c_1^T & c_2^T \end{bmatrix} \\ &= \begin{bmatrix} e_1^T & e_2^T & e_3^T + m^T\lfloor q/2 \rceil \end{bmatrix} \cdot \begin{bmatrix} Q & P \\ I_{n_2} & 0 \\ 0 & I_l \end{bmatrix} \\ &\in \mathbb{Z}_q^{1\times(n_2+l)}. \end{aligned} \tag{3}$$

- $Dec(c^T, R_2)$: output $decode(c_1^T R_2 + c_2^T) \in \{0,1\}^l$, where decode($x$) returns 0 if $|x| < \lfloor q/4 \rceil$, else returns 1:

$$c_1^T R_2 + c_2^T = \begin{bmatrix} c_1^T & c_2^T \end{bmatrix} \cdot \begin{bmatrix} R_2 \\ I \end{bmatrix} = e^T R + m^T\lfloor q/2 \rceil, \tag{4}$$

where $R = \begin{bmatrix} R_1 \\ R_2 \\ I \end{bmatrix}$ and $e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$.

The decryption will be correct as long as the absolute value of each entry of $e^T R$ is smaller than $\lfloor q/4 \rceil$.

2.4. Secret sharing

In a secret sharing scheme, the goal is to share a secret among a set of parties, called participants, denoted by $\mathcal{P}$. A trusted third party, named dealer, assigns a private value, called share, to each participant.

Only the authorized subsets of participants can recover the secret by running a pre-specified algorithm. The set of all authorized subsets is called an access structure. In general, an access structure is a subset of the power set of $\mathcal{P}$. A specific instance of general access structure is the threshold structure, which, for given t, consists of all subsets of at least t elements of the power set of $\mathcal{P}$.

A secret sharing scheme usually consists of two phases:

- Share distribution: In this phase, the shares are computed by a dealer using a pre-specified algorithm and are sent securely to the participants;
- Secret reconstruction: In this phase, the authorized subset of participants send their shares to a combiner to obtain the secret by running the algorithm.

The above-defined protocol is simple and cannot be used in real application directly. Depending on the features, TSSSs can be extended as follows:

- *Verifiable secret sharing:* In a verifiable secret sharing scheme, the dealer commits the distributed shares to each participant, and the participants can verify the validity of the recovered secrets by the combiner [19];
- *Multi-secret sharing:* In a multi-secret sharing scheme, more than one secret is shared among the participants, and it is desirable to give the participants only one share for recovering all the secrets [13];
- *Multi-stage secret sharing:* A multi-stage secret sharing scheme is a special case of multi-secret sharing schemes in which the secrets can be recovered at different stages, and the reconstructed secrets do not leak any information about the unrecovered secrets [18,43];

- *Multi-use secret sharing:* For sharing a new set of secrets, the old shares and the old public information can be used, such that sending the new shares to the participants through a secure channel is not required [18,43];
- *Changeable threshold secret sharing:* The scheme is capable of increasing the threshold in such a way that resharing the secret is not necessary by using the new threshold.

2.5. Threshold multi-stage secret sharing scheme

In this section, we introduce the lattice-based (t, n) TMSSS scheme [36], where t participants are required to recover each of the secrets. This scheme enables the participants to recover each secret independently, and it is computationally difficult to use them for obtaining any information about unrecovered secrets. In this scheme, there are m secrets $\mathrm{s}_i \in \mathbb{Z}_q^t$, $i = 1, \cdots, m$, where q is a prime number and t is the threshold. The dealer randomly selects vector $v \in \mathbb{Z}_q^t$ and publishes it. Then, for each secret s_i, he finds private lattice-basis B_i, such that:

$$\mathrm{s}_i = B_i v, \qquad i = 1, \cdots, m, \tag{5}$$

where $B_i \in \mathbb{Z}_q^{t \times t}$ is a basis for the t-dimensional lattice.

After computing the private lattice basis, B_i, $i = 1, \cdots, m$, the dealer chooses n public vectors $\lambda_j \in \mathbb{Z}_q^t$, $j = 1, \cdots, n$, such that every t of these vectors is linearly independent. Then, the dealer must find public matrices, $A_i \in \mathbb{Z}_q^{t \times r}$, $i = 1, \cdots, m$, and private shares $s_j \in \{0,1\}^r$, $j = 1, \cdots, n$, such that equality $A_i s_j = B_i \lambda_j$ holds for $i = 1, \cdots, m$ and $j = 1, \cdots, n$, where $r \geq \max(t \log t, n)$. Hence, the dealer first randomly chooses n items of shares s_j from $\{0,1\}^r$, and then solves a system of linear equations to find matrices $A_i, i = 1, \cdots, m$ for each secret.

For verification of the shares by the participants, the dealer chooses random matrix $F \in \mathbb{Z}_q^{t \times r}$ and publishes it along with the hash values of the shares as the vectors of $h_j = F s_j$, $j = 1, \cdots, n$. In addition, the dealer publishes $H(\mathrm{s}_i)$, $i = 1, \cdots, m$ for verification of the recovered secrets by the participants, where $H(\cdot)$ is a public hash function.

Distribution phase

- The dealer distributes share vector, s_j, to participant P_j through a secure channel and publishes matrices A_i, $i = 1, \cdots, m$, and vectors λ_j, $j = 1, \cdots, n$, on the bulletin board;
- Participant P_j verifies whether the hash value of the received share from the dealer is the same as that on the bulletin board, i.e. $F s_j \stackrel{?}{=} h_j$.

Combination phase

Here, different secrets are reconstructed independently. Suppose that subset $\{j_1, \cdots, j_t\} \subseteq \{1, \cdots, n\}$ of the participants intend to recover secret s_i, $i \in \{1, \cdots, m\}$. For this purpose, participant j_l, $l = 1, \cdots, t$, computes vector $d_{j_l}^i = A_i s_{j_l}$, $l = 1, \cdots, t$, as his pseudo-secret share and sends the result to the combiner in a secure manner. The combiner constructs matrix $D_i = \begin{bmatrix} d_{j_1} & \cdots & d_{j_t} \end{bmatrix}$, and then recovers secret s_i by computing $\mathrm{s}_i = D_i \Lambda^{-1} v$, where $\Lambda = \begin{bmatrix} \lambda_{j_1} & \cdots & \lambda_{j_t} \end{bmatrix}$. The participants can verify the recovered secret using the corresponding hash value, $H(\mathrm{s}_i)$, published on the bulletin board.

3. Threshold increasing algorithm

In this section, we improve our previously introduced lattice-based (t, n) TMSSS scheme [36] by proposing an algorithm for increasing the threshold from t to $t' > t$. It should be noted that a trusted party, who might not know the secret, runs the threshold increase protocol. For the realization of this purpose, we use the zero addition protocol [44], in which we share the zero secret among the participants using the new threshold, t', and combine the temporary results with the corresponding parameters of the (t, n) TMSSS scheme in such a way that the new scheme is (t', n) TMSSS scheme. The algorithm consists of the following phases.

Phase 1. Extending the dimensions from t to t'

First, we increase the size of the arrays of parameters of the original scheme from t to t' in such a way that the original equations remain correct.

$$\begin{aligned} \mathrm{s}_{i_{t\times 1}} = B_{i_{t\times t}} v_{t\times 1} &\Rightarrow \begin{bmatrix} \mathrm{s}_i \\ 0_{(t'-t)\times 1} \end{bmatrix}_{t'\times 1} \\ &= \begin{bmatrix} B_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ v'' \end{bmatrix}_{t'\times 1}, \\ A_{i_{t\times r}} s_{j_{r\times 1}} = B_{i_{t\times t}} \lambda_{j_{t\times 1}} &\Rightarrow \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}_{t'\times r'} \begin{bmatrix} s_j \\ s_j'' \end{bmatrix}_{r'\times 1} \\ &= \begin{bmatrix} B_i & 0 \\ 0 & 0 \end{bmatrix}_{t'\times t'} \begin{bmatrix} \lambda_j \\ \lambda_j'' \end{bmatrix}_{t'\times 1}, \end{aligned} \tag{6}$$

where v'' and λ_j'', $j = 1, \cdots, n$, are chosen randomly from $\mathbb{Z}_q^{(t'-t)\times 1}$ and s_j'', $j = 1, \cdots, n$, is chosen randomly from $\{0,1\}^{(r'-r)\times 1}$, where $r' = \max(t' \log t', n)$.

Phase 2. Sharing zero secret

Now, we share the zero secret according to the original scheme using the new threshold t'. Here, the difference is that vectors v, v'', s_j, s_j'', λ_j, and λ_j'' are the same as those used in the current (t, n) TMSSS scheme, i.e. Eq. (6):

$$0_{t'\times 1} = B''_{t'\times 1}\begin{bmatrix} v \\ v'' \end{bmatrix}_{t'\times 1},$$

$$A''_{t'\times r'}\begin{bmatrix} s_j \\ s''_j \end{bmatrix}_{r'\times 1} = B''_{t'\times t'}\begin{bmatrix} \lambda_j \\ \lambda''_j \end{bmatrix}_{t'\times 1}, \qquad j=1,\cdots,n, \quad (7)$$

where B'' is obtained from the first equation, and then A'' is obtained from the second equation in Eq. (7) using computed B''.

Phase 3. Parameter combination

By adding the corresponding equations of (6) and (7) together, Eq. (8) is obtained as follows:

$$\begin{bmatrix} \mathrm{s}_i \\ 0_{(t'-t)\times 1} \end{bmatrix}_{t'\times 1} = \left(B'' + \begin{bmatrix} B_i & 0 \\ 0 & 0 \end{bmatrix}\right)_{t'\times t'} \begin{bmatrix} v \\ v'' \end{bmatrix}_{t'\times 1},$$

$$\left(A'' + \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}\right)_{t'\times r'} \begin{bmatrix} s_j \\ s''_j \end{bmatrix}_{r'\times 1}$$

$$= \left(B'' + \begin{bmatrix} B_i & 0 \\ 0 & 0 \end{bmatrix}\right)_{t'\times t'} \begin{bmatrix} \lambda_j \\ \lambda''_j \end{bmatrix}_{t'\times 1}. \quad (8)$$

Let us define:

$$\mathrm{s}'_{i_{t'\times 1}} = \begin{bmatrix} \mathrm{s}_i \\ 0_{(t'-t)\times 1} \end{bmatrix}, \qquad i=1,\cdots,m,$$

$$B'_{i_{t'\times t'}} = B'' + \begin{bmatrix} B_i & 0 \\ 0 & 0 \end{bmatrix}, \qquad i=1,..,m,$$

$$v'_{t'\times 1} = \begin{bmatrix} v \\ v'' \end{bmatrix},$$

$$A'_{i_{t'\times r'}} = A'' + \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}, \qquad i=1,..,m,$$

$$s'_{j_{r'\times 1}} = \begin{bmatrix} s_j \\ s''_j \end{bmatrix}, \qquad j=1,\cdots,n,$$

$$\lambda'_{j_{t'\times 1}} = \begin{bmatrix} \lambda_j \\ \lambda''_j \end{bmatrix}, \qquad j=1,\cdots,n. \quad (9)$$

Using the new defined parameters, the following equations hold:

$$\mathrm{s}'_i = B'_i v', \qquad i=1,\cdots,m,$$

$$A'_i s'_j = B'_i \lambda'_j, \qquad i=1,\cdots,m, \quad j=1,\cdots,n. \quad (10)$$

The above equation illustrates the new TMSSS scheme using new threshold, t'. Vectors s'_j, $j=1,\cdots,n$, are the new shares and array parameters v', λ'_j, $j=1,\cdots,n$, and A'_i, $i=1,\cdots,m$, are the public information published on the bulletin board. ■

It is worth mentioning that all features of the original scheme, such as verifiability, being multi-use and multi-stage, remain unchanged under the threshold increase.

4. Threshold decryption algorithm

In this section, we introduce a threshold decryption algorithm for LWE-based public key encryption scheme of Lindner and Peikert, described in Section 2.3. The algorithm consists of the following phases:

Phase 1. Sharing the private key

First, we share the private key by sharing columns of R_2, i.e. r_2^i, $i=1,\cdots,l$, according to [36], described in Section 2.5., as follows. Let $n_1 = n_2 = t$, where t is the threshold.

$$r_2^i = B_i v, \qquad i=1,\cdots,l,$$

$$A_i s_j = B_i \lambda_j, \qquad i=1,\cdots,l, \quad j=1,\cdots,n, \quad (11)$$

where vectors s_j, $j=1,\cdots,n$, are the shares, sent securely to the participants. Parameters A_i, $i=1,\cdots,l$, and v are published on the bulletin board.

Phase 2. Threshold decryption

$TDec(c^T, s_{j_1},\cdots,s_{j_t})$: To decrypt ciphertext $c^T = \begin{bmatrix} c_1^T & c_2^T \end{bmatrix}$ using t shares $s_{j_1},\cdots,s_{j_t}$, each of t participants applies partial decryption to the ciphertext and sends the result to the combiner. The partial decryption for participant j_k, $k=1,\cdots,t$, is described by:

$$d_{j_k} = \begin{bmatrix} d^1_{j_k} \\ \vdots \\ d^l_{j_k} \end{bmatrix} = \begin{bmatrix} c_1^T A_1 s_{j_k} + w^1_{j_k} \\ \vdots \\ c_1^T A_l s_{j_k} + w^l_{j_k} \end{bmatrix}$$

$$= \begin{bmatrix} c_1^T B_1 \lambda_{j_k} + w^1_{j_k} \\ \vdots \\ c_1^T B_l \lambda_{j_k} + w^l_{j_k} \end{bmatrix}, \quad (12)$$

where $w^i_{j_k}$, $k=1,\cdots,t$, $i=1,\cdots,l$, is a discrete Gaussian noise chosen from $D_{\mathbb{Z},s_w}$. The combiner computes:

$$m' = \begin{bmatrix} d_{j_1} & \cdots & d_{j_t} \end{bmatrix} \Lambda^{-1} v + c_2 = R_2^T c_1 + c_2 + W\Lambda^{-1} v,$$

where:

$$\Lambda = \begin{bmatrix} \lambda_{j_1} & \cdots & \lambda_{j_t} \end{bmatrix},$$

and:

$$W = \begin{bmatrix} w^1_{j_1} & \cdots & w^1_{j_t} \\ \vdots & \vdots & \vdots \\ w^l_{j_1} & \cdots & w^l_{j_t} \end{bmatrix}.$$

According to Section 2.3., $decode(m'^T)$ outputs message m if the error and $\|W\Lambda^{-1}v\|_\infty$ are small compared to $\lfloor q/4 \rceil$. ■

5. Security analysis

In this section, we investigate the security of the proposed schemes in the standard model.

5.1. Threshold increase

In changing the threshold from t to t', we first share the zero secret using threshold t'. By exploiting the additive homomorphic property of the original TMSSS scheme, we add its parameter arrays with those of the zero secret sharing scheme and obtain a new (t', n) TMSSS scheme sharing the original secrets. Since the new TMSSS scheme is an extension of the original scheme using t' dimensional lattices, it inherits the security of the original scheme which has been proven in [36]. However, it is required that the new parameters fulfill the security requirements of the cryptographic primitives used in the new scheme.

The definition of Ajtai's one-way function $f(x) = Ax$ implies that matrix A is chosen uniformly at random. Hence, we show that matrices A'_i, $i = 1, \cdots, m$, obtained by Eq. (9) have uniform distribution on $\mathbb{Z}_q^{t' \times r'}$. From the proof of Lemma 1 in [36], we know that adding or multiplying a uniformly random matrix to/by a matrix with independently arbitrary distribution results in a matrix with a uniform distribution. Furthermore, A'', obtained when sharing the zero secret, has a uniform distribution on $\mathbb{Z}_q^{t' \times r'}$ by Lemma 1 in [36]. Therefore:

$$A'_{i_{t' \times r'}} = A'' + \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix}$$

has a uniform distribution on $\mathbb{Z}_q^{t' \times r'}$.

5.2. Threshold decryption

In this section, we analyze the security of the proposed threshold decryption algorithm for the LWE-based public key encryption scheme by Lindner and Peikert. The security of the TMSSS scheme and the above-mentioned encryption scheme are proven in Section 5.1. [31,36], respectively. In a threshold decryption algorithm, it is desired that no information about the private key is leaked, i.e. no one using the results of partial decryption d_{j_k}, $k = 1, \cdots, t$, can obtain any information about private key R_2. We prove this assertion in the following theorem:

Theorem 1. *Let $s_1, \cdots, s_n$ be the shares corresponding to private key R_2. Using:*

$$d_{j_k} = \begin{bmatrix} d_{j_k}^1 & \cdots & d_{j_k}^l \end{bmatrix}^T, \quad k = 1, \cdots, t,$$

private key R_2 cannot be revealed in polynomial time, where:

$$d_{j_k}^i = c_1^T A_i s_{j_k} + w_{j_k}^i, \quad i = 1, \cdots, l, \quad j = 1, \cdots, t.$$

Proof. Assume that $y_{j_k}^i = A_i s_{j_k}$, $i = 1, \cdots, l$ and $k = 1, \cdots, t$. Solving equation $d_{j_k}^i = c_1^T A_i s_{j_k} + w_{j_k}^i = c_1^T y_{j_k}^i + w_{j_k}^i$ with respect to $y_{j_k}^i$ leads to an LWE problem. Furthermore, solving equation $y_{j_k}^i = A_i s_{j_k}$ with respect to s_{j_k} leads to an ISIS problem. Since both problems cannot be solved in polynomial time, one cannot obtain any information about the shares in polynomial time using partial decryption $d_{j_k}^i$.

On the other hand, obtaining private key R_2 from expression $\begin{bmatrix} d_{j_1} & \cdots & d_{j_t} \end{bmatrix} \Lambda^{-1} v = R_2^T c_1 + W \Lambda^{-1} v$ computed during the threshold decryption process, leads to an LWE problem which is hard to solve in polynomial time. □

6. Performance analysis

In this section, we investigate the performance of the proposed schemes.

6.1. Threshold increase

We consider the changeable TMSSS scheme in which we deal with two phases:

1. Sharing the secret using threshold t;
2. Changing the threshold from t to t'.

In the first phase, matrices A_i, $i = 1, \cdots, m$, published on the bulletin board, are dominant from memory consumption point of view. On the other hand, the memory consumption of shares s_j, $j = 1, \cdots, n$, should be taken into account. In Table 1, the memory

Table 1. Memory requirements for different schemes.

Scheme	Size of public values per secret size	Size of each share per secret size
He & Dawson [16]	$m \times n$	1
Harn [45]	$m \times (n-t)$	1
Chang [18]	$m \times n$	1
Das [19]	$m \times t \times n$	1
Harn [46]	$\frac{m^2(n+1)}{t}$	m
The proposed TMSSS scheme	$m \times r$	$r/(t \log q)$

requirements for A_i, $i = 1, \cdots, m$, and s_j, $j = 1, \cdots, n$, per secret size, i.e. $\log_2(q)$, are given. Table 1 illustrates that in the proposed scheme, the size of each share per secret size is $\frac{r}{t \log q}$, which equals 0.5 if we let $q = t^2$ and $r = t \log t$. In the second phase, matrices A_i, $i = 1, \cdots, m$, are changed to A'_i, $i = 1, \cdots, m$, where the size of public values per secret size is changed to $m \times r' \times \frac{t'}{t}$. When increasing the threshold from t to t', shares s_j, $j = 1, \cdots, n$, are changed to:

$$s'_j = \begin{bmatrix} s_j \\ s''_j \end{bmatrix}, \quad j = 1, \cdots, n.$$

In this case, we only need to send vectors s''_j, $j = 1, \cdots, n$, to the participants. In this way, the protocol needs less data communication to the participants over a secure channel.

From complexity's point of view, in the share distribution phase of the (t, n) threshold scheme, the computational complexity of public matrices A_i, $i = 1, \cdots, m$, is of $O(t^2 n) + O(tn(r-n)) + O(n^3) + O(tn^2) \sim O(n^3)$ for each secret and consists of three matrix multiplications and one matrix inversion [36]. Each secret recovery consists of two steps:

1. The participants' side: In computing the pseudo-secret shares, since the shares are binary arrays, computing the pseudo-secret shares only requires simple column addition in matrix A_i, which has the complexity of $O(tr)$ for each participant. This makes the scheme suitable for the applications with low complexity requirements on the participants' side;
2. The combiner's side: This step has the complexity of $O(t^3)$, which consists of one matrix multiplication and one matrix inversion.

In changing the threshold from t to t', the dominant part is sharing the zero secret among the participants, which is of $O(t'^2 n) + O(t' n(r' - n')) + O(n^3) + O(t' n^2) \sim O(n^3)$ for the zero secret and consists of three matrix multiplications and one matrix inversion. The remaining parts only use additions which can be ignored when compared to the zero secret sharing.

6.2. Threshold decryption algorithm

In view of computational complexity, the proposed threshold decryption algorithm consists of two parts:

1. The partial decryption by the participants has the computational complexity of $O(lt^2 r)$;
2. The final decryption by the combiner using the output of the partial decryption has the complexity of $O(lt^2)$.

Both parts only use matrix operations which are more efficient than the exponentiations used in the traditional schemes.

7. Conclusions

In a TSSS, when the secret is threatened by some corrupted participants, or the organizational structure of the participants is to be changed, it might be required that the threshold be increased. In this paper, we have proposed a threshold increasing algorithm for our previously proposed lattice-based (t, n) TMSSS scheme which supports the threshold changeability feature, in addition to the inherited features of being multi-stage, multi-use, and verifiable. For realization of the new feature, we share the zero secret with the new threshold and combine the new array parameters with those of the original scheme. Furthermore, based on Lindner and Peikert's public key encryption scheme, we have introduced a lattice-based threshold decryption algorithm using the improved TMSSS scheme. For threshold decryption, the participants partially decrypt the ciphertext using their shares and send the results to the combiner, who can then decrypt the ciphertext using the partial decryption results. We have discussed the security of both schemes based on the hardness of lattice problems, i.e. the LWE and ISIS problems, which are believed to resist against the quantum algorithms. Moreover, both schemes are efficient, especially in the participants' side, because of simple matrix operations used in the computations of the changeable TMSSS and threshold decryption protocols. Hence, they are suitable for the mobile applications, such as smart cards and sensor networks, in which low processing capability of the used devices is a dominant factor.

References

1. Chunying, W., Shundong, L. and Yiying, Z. "Key management scheme based on secret sharing for wireless sensor network", In *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*, pp. 574-578 (2013).
2. Chang, C.-C., Chen, W.-Y. and Chang, S.-C. "A highly efficient and secure electronic cash system based on secure sharing in cloud environment", *Security and Communication Networks*, **9**(14), pp. 2476-2483 (2016).
3. Yuan, L., Li, M., Guo, C., Hu, W. and Tan, X. "A verifiable E-voting scheme with secret sharing", In *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, pp. 304-308 (2015).
4. Kanai, A., Tanimoto, S. and Sato, H. "Performance evaluation on data management approach for multiple clouds using secret sharing scheme", In *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 471-473 (2016).
5. Shamir, A. "How to share a secret", *Commun. ACM*, **22**(11), pp. 612-613 (1979).

6. Blakley, G.R. "Safeguarding cryptographic keys", In *Proceedings of the 1979 AFIPS National Computer Conference*, **48**, pp. 313-317 (1979).
7. Stadler, M. "Publicly verifiable secret sharing", In *Advances in Cryptology EUROCRYPT 96*, U. Maurer, Ed., LNCS 1070, pp. 190-199, Springer Berlin Heidelberg (1996).
8. Barletta, A., Callegari, C., Giordano, S., Pagano, M. and Procissi, G. "Privacy preserving smart grid communications by verifiable secret key sharing", In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pp. 199-204 (2015).
9. Brickell, E. and Stinson, D. "The detection of cheaters in threshold schemes", In *Advances in Cryptology CRYPTO 88*, S. Goldwasser, Ed., LNCS 403, pp. 564-577, Springer New York (1990).
10. Mahmoud, Q.A. "A novel verifiable secret sharing with detection and identification of cheaters' group", *International Journal of Mathematical Sciences and Computing (IJMSC)*, **2**(2), pp. 1-13 (2016).
11. Martin, K., Safavi-Naini, R. and Wang, H. "Bounds and techniques for efficient redistribution of secret shares to new access structures", *The Computer Journal*, **42**(8), pp. 638-649 (1999).
12. Barwick, S.G., Jackson, W.-A. and Martin, K. "Updating the parameters of a threshold scheme by minimal broadcast", *Information Theory, IEEE Transactions on*, **51**(2), pp. 620-633 (2005).
13. Blundo, C., De Santis, A., Di Crescenzo, G., Gaggia, A.G. and Vaccaro, U. "Multi-secret sharing schemes", In *Advances in Cryptology CRYPTO94*, LNCS 839, pp. 150-163 (1994).
14. Fatemi, M., Ghasemi, R., Eghlidos, T. and Aref, M. "Efficient multistage secret sharing scheme using bilinear map", *Information Security, IET*, **8**(4), pp. 224-229 (2014).
15. Pang, L., Li, H. and Wang, Y. "An efficient and secure multi-secret sharing scheme with general access structures", *Wuhan University Journal of Natural Sciences*, **11**(6), pp. 1649-1652 (2006).
16. He, J. and Dawson, E. "Multistage secret sharing based on one-way function", *Electronics Letters*, **30**(19), pp. 1591-1592 (1994).
17. Geng, Y.J., Fan, X.H. and Fan, H. "A new multi-secret sharing scheme with multi-policy", In *Advanced Communication Technology, the 9th International Conference on*, **3**, pp. 1515-1517 (2007).
18. Chang, T.Y., Hwang, M.S. and Yang, W.P. "A new multi-stage secret sharing scheme using one-way function", *SIGOPS Oper. Syst. Rev.*, **39**(1), pp. 48-55 (2005).
19. Das, A. and Adhikari, A. "An efficient multi-use multi-secret sharing scheme based on hash function", *Applied Mathematics Letters*, **23**(9), pp. 993-996 (2010).
20. Chien, H.-Y., JAN, J.-K. and Tseng, Y.-M. "A practical (t, n) multi-secret sharing scheme", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **83**(12), pp. 2762-2765 (2000).
21. Yang, C.-C., Chang, T.-Y. and Hwang, M.-S. "A (t, n) multi-secret sharing scheme", *Applied Mathematics and Computation*, **151**(2), pp. 483-490 (2004).
22. Shao, J. and Cao, Z. "A new efficient (t, n) verifiable multi-secret sharing (VMSS) based on YCH scheme", *Applied Mathematics and Computation*, **168**(1), pp. 135-140 (2005).
23. Steinfeld, R., Pieprzyk, J. and Wang, H. "Lattice-based threshold changeability for standard Shamir secret-sharing schemes", *Information Theory, IEEE Transactions on*, **53**(7), pp. 2542-2559 (2007).
24. Zhang, Z., Chee, Y.M., Ling, S., Liu, M. and Wang, H. "Threshold changeable secret sharing schemes revisited", *Theoretical Computer Science*, **418**, pp. 106-115 (2012).
25. Lou, T. and Tartary, C. "Analysis and design of multiple threshold changeable secret sharing schemes", In *Cryptology and Network Security*, M. Franklin, L. Hui and D. Wong, Eds., LNCS 5339, pp. 196-213. Springer Berlin Heidelberg (2008).
26. Steinfeld, R., Pieprzyk, J. and Wang, H. "Lattice-based threshold-changeability for standard CRT secret-sharing schemes", *Finite Fields and Their Applications*, **12**(4), pp. 653-680, Special Issue Celebrating Prof. Zhe-Xian Wan's 80th Birthday (2006).
27. Shor, P.W. "Algorithms for quantum computation: Discrete logarithms and factoring", In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, pp. 124-134, Washington, DC, USA IEEE Computer Society (1994).
28. Ajtai, M. "Generating hard instances of lattice problems (extended abstract)", In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pp. 99-108, New York, NY, USA (1996). ACM.
29. Micciancio, D. and Goldwasser, S., *Complexity of Lattice Problems: A Cryptographic Perspective*, Milken Institute Series on Financial Innovation and Economic Growth, Springer US (2002). ISBN: 978-1-4613-5293-8 (Print), 978-1-4615-0897-7 (Online).
30. Regev, O. "On lattices, learning with errors, random linear codes and cryptography", In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC '05*, pp. 84-93, New York, NY, USA ACM (2005).
31. Lindner, R. and Peikert, C. "Better key sizes (and attacks) for LWE-based encryption", In *Topics in Cryptology CT-RSA 2011*, A. Kiayias, Ed., LNCS 6558, pp. 319-339, Springer Berlin Heidelberg (2011).
32. Georgescu, A. "A LWE-based secret sharing scheme", *IJCA Special Issue on Network Security and Cryptography*, **NSC**(3), pp. 27-29, Published by Foundation of Computer Science, New York, USA (2011).

33. El Bansarkhani, R. and Meziani, M. "An efficient lattice-based secret sharing construction", In *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*, LNCS 7322, pp. 160-168, Springer Berlin Heidelberg (2012).

34. Khorasgani, H.A., Asaad, S., Eghlidos, T. and Aref, M. "A lattice-based threshold secret sharing scheme", In *Information Security and Cryptology (ISCISC), 2014 11th International ISC Conference on*, pp. 173-179 (2014).

35. Asaad, S., Khorasgani, H.A., Eghlidos, T. and Aref, M. "Sharing secret using lattice construction", In *Telecommunications (IST), 2014 7th International Symposium on*, pp. 901-906 (2014).

36. Pilaram, H. and Eghlidos, T. "An efficient lattice based multi-stage secret sharing scheme", *Dependable and Secure Computing, IEEE Transactions on*, PrePrint (2015).

37. Bendlin, R. and Damgård, I. "Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems", In *Theory of Cryptography*, LNCS 5978, pp. 201-218, Springer (2010).

38. Frederiksen, T.K. "A multi-bit threshold variant of Regev's LWE-based cryptosystem", *Cryptology ePrint Archive* (2011).

39. Singh, K., Pandu Rangan, C. and Banerjee, A. "Lattice based efficient threshold public key encryption scheme", *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, **4**(4), pp. 93-107 (2013).

40. Bendlin, R., Krehbiel, S. and Peikert, C. "How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE", In *Applied Cryptography and Network Security*, LNCS 7954, pp. 218-236 (2013).

41. Bernstein, D., Buchmann, J. and Dahmen, E., *Post-Quantum Cryptography*, Springer (2009).

42. Gentry, C., Peikert, C. and Vaikuntanathan, V. "Trapdoors for hard lattices and new cryptographic constructions", In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, ACM, pp. 197-206 (2008).

43. Chang, T.-Y., Hwang, M.-S. and Yang, W.-P. "An improved multi-stage secret sharing scheme based on the factorization problem", *Information Technology and Control*, **40**(3), pp. 246-251 (2011).

44. Nojoumian, M. and Stinson, D.R. "On dealer-free dynamic threshold schemes", *Advances in Mathematics of Communications*, **7**(1), pp. 39-56 (2013).

45. Harn, L. "Comment on multistage secret sharing based on one-way function", *Electronics Letters*, **31**(4), pp. 262-268 (1995).

46. Harn, L. "Secure secret reconstruction and multi-secret sharing schemes with unconditional security", *Security and Communication Networks*, **7**(3), pp. 567-573 (2014).

Acknowledgement

This work was supported by Iran National Science Foundation (INSF) under grant number 94017742.

Biographies

Hossein Pilaram received his BSc degree in Electrical Engineering and the MSc degree in Communication Systems from the Sharif University of Technology, Tehran, Iran, in 2010 and 2012, respectively. He is currently working on his PhD dissertation in the Department of Electrical Engineering of Sharif University of Technology. His research interests are cryptography, coding theory, and mobile networks.

Taraneh Eghlidos received her BSc degree in Mathematics in 1986 from the University of Shahid Beheshti, Tehran, Iran, and the MSc degree in Industrial Mathematics in 1991 from the University of Kaiserslautern, Germany. She received her PhD degree in Mathematics in 2000, from the University of Giessen, Germany. She joined the Sharif University of Technology (SUT) in 2002, and is currently an Associate Professor in the Electronics Research Institute of SUT. Her research interests include interdisciplinary research areas such as symmetric and asymmetric cryptography, application of coding theory in cryptography, and mathematical modeling for representing and solving real world problems. Her current research interests include lattice-based cryptography and code-based cryptography.