A Security Definition for Multi Secret Sharing and a Scheme Based on LWE

Massoud Hadian Dehkordi, Iran University of Science and Technology
Reza Ghasemi, Iran University of Science and Technology

Since the advent of secret sharing scheme many researches have been allocated to study on this topic because it has a lot of application. For the first time Shamir and Blakley introduced the concepts of secret sharing. In their scheme, just one secret is shared. After a while, Harn present a scheme in which many secrets can be shared, but the secrets have to recover in predetermined order. In addition, in his scheme just one share is assigned to each participant. After a while, many scheme introduced such that they have not any constraint on the order of recovering secret. These kind of scheme is called Multi secret sharing scheme and it abbreviated by MSS. To the best of our knowledge, up until now, no exact definition for the security of MSS scheme has been presented. In this paper, a definition for secrecy of MSS scheme is introduced and a MSS scheme is present based on Learning With Error (LWE). LWE is a one of lattice concepts which nowadays constitutes the core of many cryptographic constructions because the hardness of lattice problems is well studied and these constructions can be reduced to NP-Hard problems. The advantage of using LWE is twofold, first is that the hardness of LWE is well understood, second working with it is very simple because just simple operations are used. At the end of the paper a verifiable version of presented MSS scheme is given. Verifiability is an important feature which has defined. In this kind of schemes, dishonest dealer or participants can be identified.

Additional Key Words and Phrases: Secret sharing, Multi Secret, Lattice, LWE, Verifiability.

1. INTRODUCTION

Secret sharing is defined as a method to share a secret between many participants such that the authorized subset of them can recover the secret by submitting their shares. In this process each of the participants is given a private information which is called share or private share. The set of the subsets of authorized participants is called an access structure and is denoted by Γ. If all of the elements which belong to an access structure have cardinal t, then we call this scheme a $(t, n)-$threshold secret sharing scheme. Secret sharing was introduced by Shamir [Shamir 1979] and Blakeley [Blakley 1899] independently. Shamir presented a $(t, n)-$threshold scheme base on interpolation. Shamir's scheme was perfect. That is, any $t-1$ participant cannot obtain any information about the secret (in view of information theory). This definition of security was relaxed and other authors introduced schemes that are computationally secure. His scheme was simple and efficient, therefore many other authors follow his way. Secret sharing plays an important role in many cryptographic protocols such as Multi-Party Protocols [Yao 1982], hence many researchers were motivated to work in this area [Harn 1995a], [Harn 1995b].

After the schemes that share just one secret, Harn present a scheme which is called Multi Secret Sharing Scheme (Abbreviated by MSS), in which many secrets are shared while just one share is assigned to each participants [Harn 1995a]. In his scheme some public values are publishes by the dealer. These public values are used in recovering the secrets. Later, the other authors introduced MSS schemes that have lesser number of public values [Chang et al. 2005], [He and Dawson 1995a]. Many authors have worked on Harn's scheme and improved his scheme by adding new features. One of the important improvement is verifiability. In verifiable schemes, participant's deception

Author's addresses: M. Hadian Dehkordi and R. Ghasemi, The School of Mathematics, Iran University of Science and Technology, Tehran, Iran
mhadian AT iust DOT ac DOT ir, ghasemi DOT basu AT gmail DOT com

can be identified [Chor et al. 1985], [Stadler 1996]. After these papers, verifiability has become an important part of secret sharing schemes.

In this paper we define a security notation for MSS scheme, then lattice conception will be used to introduce a new threshold MSS scheme which satisfies presented security definition. Informally, lattice is a discrete subgroup of $\mathbb{R}^n$ or equivalently integer combination of a few independent vectors in $\mathbb{R}^n$. Many computational problems are related to the lattice conception [Micciancio and Regev 2009], [Regev 2006], e.g. finding shortest vector problem (SVP), closest vector problem (CVP), shortest independent vector problem $(SIVP)$, Learning With Error (LWE) and many other problems. This problems are believed to be hard and to the best of our knowledge the best algorithm for solving them need exponential time. For instance, it is proved that CVP is a NP-Hard problem. Therefore, until $NP \neq P$ no one can solve CVP problem in polynomial time. In other word, cryptographic constructions which are formed base on CVP can not be broken in polynomial time until $NP = P$ holds. These specifications have caused using them widely in new cryptographic constructions [Kawachi et al. 2007], [Agrawal et al. 2010], [Akavia et al. 2009]. In addition, these constructions, seemingly, are faster in comparison with other kind of constructions which are not based on lattice because mathematical operations, which are used in lattice, are very simple.

Mentioned feature of lattice theory motivated researchers to use lattice in secret sharing scheme [El Bansarkhani and Meziani 2012], [Steinfeld et al. 2004], therefore in this paper, we present an MSS scheme and verifiable version of it based on lattice conceptions. To the best of our knowledge, our scheme has a minimum number of public values in all MSS schemes and inheritance good feature of lattice such as simplicity, fastness, security and so on.

The paper is structured as follows: in section 2 we introduce lattice concepts that are needed in next sections, then in section 3 a security definition for MSS schemes is define and an MSS scheme is presented in 4 and we prove that it satisfies the presented security. Next section is dedicated to presenting verifiable version of presented MSS scheme. Finally, the last section is conclusion.

2. PRELIMINARIES

In this section we review some concepts and introduce some notations which are needed in the rest of this paper. The following notations are commonly used in this paper, hence they are introduced. The notation $\in_r$ means choosing uniformly from a finite set. For two vectors $A = [a_1, \ldots, a_n]$ and $B = [b_1, \ldots, b_n]$ we define $\langle A.B \rangle = \sum_{i=1}^{n} a_i b_i$ and sometimes for simplicity instead of $\langle A.B \rangle$, we just write $A.B$ or AB. The notation g^A refers to the value $[g^{a_1}, \ldots, g^{a_n}]$. we will use the rot function in this paper a lot. It is defined as follows,

$$rot^j(A) = [a_{j+1}, a_{j+2}, \ldots, a_n, a_1, \ldots, a_j]$$

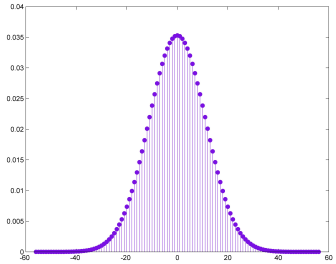
At rest of this paper, $\overline{1}_n$ is used for showing the vector $[1, 1, \ldots, 1]_{1\times n}$. The notation $\tilde{O}$ is a variant of big O notation in which ignores logarithmic factors. A distribution which may be new for readers is ψ_α distribution over $\mathbb{Z}_q$. This distribution defines as normal distribution rounded to the nearest integer with mean zero and standard deviation αq. For instance, the following figure shows the ψ distribution with $\alpha = 0.1$ over $\mathbb{Z}_{113}$,

Another important distribution is $\mathcal{A}_{s,\chi}$. For a specific $s \in \mathbb{Z}_q^n$, distribution χ induces a distribution over the pairs (A, V) where V has a form of $\langle A.s \rangle + e$ in which A is chosen from $\mathbb{Z}_q^n$ uniformly and e is chosen according χ distribution.

We talked about lattice before. Here we mention the exact definition of lattice;

Definition 2.1. Suppose $b_1, b_2, \ldots, b_m$ are m linearly independent vectors in $\mathbb{R}^n$ $(m \leq n)$, therefore the lattice that generated by these vectors is linear integer combination

Fig. 1. ψ distribution



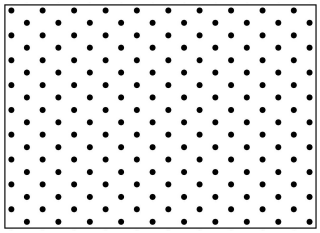
of them,

$$\mathcal{L}(b_1, b_2, \ldots, b_m) = \{ \sum_{i=1}^{m} z_i b_i | z_i \in \mathbb{Z} \}$$

The vectors $b_1, b_2, \ldots, b_m$ are called a basis for the lattice. Basis is not unique and it can be shown that B' is another basis for $\mathcal{L}(B)$ if and only if $B' = BU$ where U is an appropriate unimodular matrix.

For example, following figure depicts a lattice that constructed from two vectors in $\mathbb{R}^2$ space.

Fig. 2. Lattice.



Many problem such as shortest vector problem (SVP), closest vector problem (CVP) , learning with errors (LWE) and the other well-known problems [Micciancio and Goldwasser 2002] play an important role in lattice theory. Many version of this problems are assessed. One of these versions is approximation version. In approximation version we have a function f along with the principal problem and the aim is to find an answer which its norm is at most f times bigger that the exact answer. For instances, in approximation problem SVP_{n^2} our goal is to find a vector in lattice which has norm at most n^2 time bigger that the shortest vector in lattice.

We introduce LWE concept in more details because we take advantage of it in designing a threshold MSS scheme. LWE is solving a system of linear equation of n variable over a field like $\mathbb{Z}_q$ in which each equation has noise. Roughly speaking, LWE

problem is solving the system of linear equation like following;

$$\begin{aligned} s_{1,1}x_1 + s_{1,2}x_2 + \ldots + s_{1,n}x_n &= y_1 + e_1 \\ s_{2,1}x_1 + s_{2,2}x_2 + \ldots + s_{2,n}x_n &= y_2 + e_2 \\ &\vdots \\ s_{m,1}x_1 + s_{m,2}x_2 + \ldots + s_{m,n}x_n &= y_m + e_m \end{aligned}$$

The values e_i, $i = 1, \ldots, m$, are noises. It has been shown that if q be a polynomial of the variable n then it is equivalent to a problem which is described as follows [Micciancio and Regev 2009];

PROBLEM 1. *Suppose n, q are integers and χ is a distribution on $\mathbb{Z}_q$. Assume that (A, v) is given $(A \in \mathbb{Z}_q^n, v \in \mathbb{Z}_q)$. Decide whether v is chosen randomly or v has form of $As + e$ that $s \in_r \mathbb{Z}_q^n$ and e is chosen according to χ distribution.*

For example, we asses this problem for the simple case $n = 2$ and $\chi = \psi_\alpha$. We have drawn two pictures. The first one denotes $A_{s,\psi}$ distribution with parameter $q = 13$ and $\alpha = 0.5$ for a specific value $s = (s_1, s_2)$. We show the probability of occurring $(A, v) = ((a_1, a_2), v)$ $(v = a_1 s_1 + a_2 s_2 + e$ where e is chosen from ψ distribution$)$ with a ball in coordinate (a_1, a_2, v) which its volume indicates the mass probability, see the figure (3). The second one is uniform distribution in which the components of (A, v) are chosen uniformly. Suppose an instances (A, v) is given, the aim is to decide this instance is chosen from which distribution.

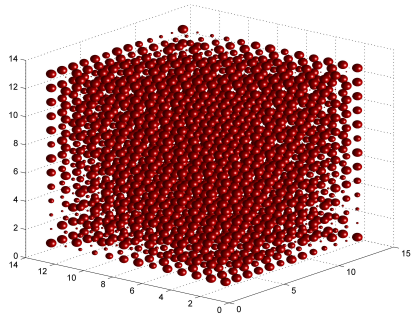


Fig. 3. $\mathcal{A}_{s,\psi}$ distribution

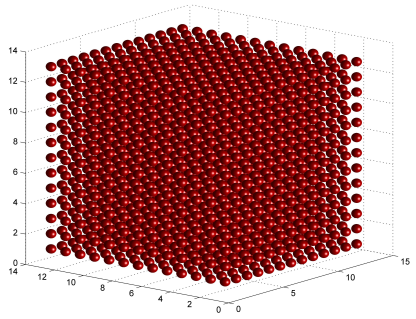


Fig. 4. $\mathcal{U}$ distribution

This problem is believed to be hard. The following theorem clarifies this claim and shows the relation between the problems $SIVP$ and LWE.

THEOREM 2.2. *[Micciancio and Regev 2009] Suppose we have access to an oracle that solves the LWE problem with parameters n, q, ψ_α where $\alpha q > \sqrt{n}$ and $q \leq poly(n)$ is prime. Then there exists a quantum algorithm that solves $SVP_{\tilde{O}(n/\alpha)}$ and $SIVP_{\tilde{O}(n/\alpha)}$ in any n-dimensional lattice.*

The above problem and theorem are the relaxed version of what is said in Micciancio et al. [Micciancio and Regev 2009] paper. Next, we assert a theorem that will be used to show computational security of presented scheme.

THEOREM 2.3. *If we access to an oracle that can computes* $(Arot^j(s)+e, j)$*, where* j *belongs to* $\{1, 2, \ldots, k-1\}$ *and* $A, s \in \mathbb{Z}_q^k$ $e \in \mathbb{Z}_q$*, given* $(A, As+e)$ *with a non-negligible probability, then we can solve the problem 1 with non-negligible probability.*

Notice that j is not constant, even we may have different j on querying the same values.

PROOF. Assume we have access to an oracle that outputs $(Arot^j(s)+e, j)$ on input $(A, As+e)$ with probability ϵ such that ϵ is a non-negligible function of k. Suppose $(A, As+e)$ is given where $A = [a_1, a_2, \ldots, a_k]$. First of all, we query $(A', As+e)$ from oracle, $A' = [0, a_1, 0, a_2, \ldots, 0, a_k]$, the output will be $(A'rot^j(s), j)$ with probability $\epsilon(2n)$. Because, in view of oracle the input looks like as follows,

$$As + e = A's' + e = [0, a_1, \ldots, 0, a_k][0, s_1, \ldots, 0, s_k] + e$$

Where $s' = [0, s_1, \ldots, 0, s_k]$. Therefore, the output of querying $(A', As+e) = (A', A's+e)$ is $(A'rot^j(s)+e, j)$ for some j. If $j \equiv 1(mod\ 2)$, then the value $A'rot^j(s)+e$ is e,

$$A'rot^j(s) + e = (\sum_{i=1}^{k} 0 \times a_i) + e = e$$

Therefore the value e can be determined with probability $\epsilon(2n)/2$. We can increase this probability by iterate this procedure. After extracting the exact value of e, solving the problem would be easy. Let's back to the main problem. Suppose the value (A, V) are chosen from $\mathcal{A}_{s,\psi}$ or $\mathcal{U}$ distribution. Our aim is to guess (A, V) is chosen from which distribution. First of all, we feed the value (A, V) to the oracle and extract e. Notice that, We can query (A, V) from the oracle more and more to ensure that the obtained value is correct. Now it can be easily check that e is chosen from distribution ψ or $\mathcal{U}$. Obviously, If e is chosen from ψ, then (A, V) is chosen from distribution $\mathcal{A}_{s,\psi}$ else V is chosen uniformly. □

On balance, above information is needed for proving security of scheme. We show that if the presented scheme be vulnerable then the problem LWE is vulnerable too. Roughly speaking, if an adversary can break our scheme in polynomial time with non-negligible probability, then it can be shown that there exist a polynomial adversary which can solve LWE problem with non-negligible probability.

3. A SECURITY DEFINITION OF MSS SCHEME

To the best of our knowledge, there is not an specific definition for secure MSS scheme until now. Requirement to an exact definition for any cryptographic construction is clear. Therefore, in this section a security definition for MSS scheme is presented and in the next sections a MSS scheme is introduced that satisfies in this definition.

Before introducing a definition, let us briefly explain the main structure of MSS schemes. Any MSS scheme consist of four main polynomial time algorithms $PaG(.)$, $Share(.)$, $Sub(.)$ and $Recover(.)$. In a MSS scheme secrets are shared step by step as follows:

— Dealer generates secret space and maximum number of secrets which can be shared regarding 1^{sp} by the algorithm $PaG(.)$ $(\mathcal{S}, m \leftarrow PaG(1^{sp}))$, sp is called security parameter.
— Secrets are chosen from the generated space ($\mathcal{S}$).
— Using the security parameter, Γ and the secrets, the participant's share and public values are determined by the algorithm $Share(.)$.
— Publishes the public values and sends participant's shares to them via a secure channel.

After finishing sharing phase, in recovering phase every authorized subset of participants can collaborate and recover any secret using their submitted sub-shares, obtained from their shares and the target secrets by the algorithm $Sub(.)$. There exist an important point here that recovering a secret should not lead to recover another unrecovered secrets. In other words, we expect that the secure MSS scheme has this feature.

After a brief introducing of MSS structure, we can define a secure MSS scheme. An MSS scheme is called secure for the family of the access structure $\mathfrak{A}$ if for any $\Gamma \in \mathfrak{A}$, it satisfies in below definition:

Definition 3.1. For any polynomial adversary $\mathcal{A}$, the probability of following event is a negligible function of security parameter:

$$\left\{ \mathcal{S}, m \leftarrow PaG(1^{sp}), (s_1, \ldots, s_m) \in_r \mathcal{S}^m \ ; \ \mathcal{A} \begin{pmatrix} a\ subset\ of \\ sub-shares \\ and\ the \\ public\ values \end{pmatrix} \in \mathcal{S}_{ur} \right\}.$$

In this definition two points should be considered. First, $\mathcal{S}_{ur}$ is the set of secrets which can not be recovered using revealed sub-shares. Second, the probability is taken over the secrets and the random bit of algorithm $\mathcal{A}$, $PaG(.)$.

Definition 3.2. We say that an algorithm can break the MSS scheme if the above experience holds for it with non-negligible probability.

Definition 3.3. If we have a secure MSS scheme with $\mathfrak{A}$ such that $\mathfrak{A}$ is consist of all threshold access structures, then we call this scheme a secure threshold MSS scheme.

Before continuing, let's see: Why we define the security notation of MSS scheme like this. Every security definition depends on our expectations of MSS scheme and the power of adversaries. These two fact affect any effort to define a security definition. Therefore, before presenting a definition we list what we expect and the power of adversaries, which the scheme should resist on their attacks. Almost in all modern cryptographic construction, the adversary is considered a probabilistic polynomial time algorithm. Here, we have follow this traditional law and considered that the scheme resist against the probabilistic polynomial time adversaries. Let's explain the worst situation which helps us to recognize the expectation better. The worst situation is when some secrets is recovered and the sub-shares related to them is revealed beside a subset of malicious participant, which this subset of participants do not belong to the Γ, collaborate to recover another unreconstructed secret, here we expect the scheme should be resist and the unrecovered secrets can not be recovered. If we wrap up these lines and express them as a mathematics terminology, we will reach to what is defined before in the definition 3.1.

We have not impose any limitation on the size of the share, therefore this definition of security for threshold MSS schemes can be easily obtained by the following scheme. We call it toy example. Suppose t, n and 1^{sp} are given. It should be introduced how the four algorithms work.

Share Distribution. The algorithm $PaG(.)$ on input 1^{sp} outputs a random natural number m and $\mathbb{Z}_q$ as a group where the secrets are chosen such that q is a arbitrary prime number and $\log_2(q) \geq sp$. After choosing $\mathbb{Z}_q$, the secrets $s_1, \ldots, s_r$ are chosen from $\mathbb{Z}_q^r$ $(r \leq m)$. The value $(s_1, \ldots, s_r, t, n)$ is given to the algorithm $Share(.)$ then, it sends $(Q_1(i), \ldots, Q_r(i))$ to the i^{th} participant as his/her share where $Q_k(x)$ is a random polynomial of degree $t-1$ in $\mathbb{Z}_q[x]$ such that $Q_k(0) = s_k$.

Secret Reconstruction. Suppose t participants $P_{i_1}, \ldots, P_{i_t}$ collaborate to reconstruct s_k therefore, each of them submit his/her sub-share whom is computed by the algorithm $Sub(.)$. This algorithm takes $(Q_1(i), \ldots, Q_r(i), k)$ and outputs $Q_k(i)$. At the end of this process they can reconstruct s_k using interpolation.

Obviously, it can be checked that this scheme satisfies the presented security definition even if we eliminate the limitation on the power of adversary. This scheme is very simple and can be used without afraid of any adversaries attacks. This scheme satisfies in presented security definitions, but at the price of assigning a large share to the participants. This can make this scheme impractical because handling and managing these big shares is hard, especially, when the number of secrets increases. With respect to this issues, many MSS schemes have presented in which many secrets can be shared and the private shares are small [He and Dawson 1995b]. In the next section, we introduce an MSS scheme based on lattice conceptions and prove that it satisfies in the presented security definition, while it benefits from small private shares.

4. AN MSSS SCHEME

As discussed above, many schemes are presented to reduce the number of public values. We use the lattice conception to introduce a threshold MSS scheme that has lower public values.

4.1. Share Distribution

Suppose sp, which is the security parameter, is given. Let q and α are a prime number and a positive real number respectively such that $\alpha q < \sqrt{sp}$ and $2 \leq q < poly(sp)$. The algorithm $PaG(.)$ works as this way. The value 1^{sp} is feeded to $PaG(.)$, then the output will be m ($m < \log sp$) and $\mathbb{Z}_q^{sp+1}$ which m denotes the maximum number of secrets that can be shared. Dealer shares m secrets $S_1, S_2, \ldots, S_m \in \mathbb{Z}_q^{sp+1}$ among n participants $P_1, \ldots, P_n$ in such a way that every t ($t \leq n$) participant can recover the secrets in an unordered manner. Dealer apply the algorithm $Share(.)$ and computes the private shares and public values. The algorithm $Share(.)$ works as follows:

(1) The sp polynomials $Q_1(x), \ldots, Q_{sp}(x) \in_r \mathbb{Z}_q[x]$ of degree $t-1$ are chosen.
(2) Chooses $e_1, e_2, \ldots, e_t \in \mathbb{Z}_q$ according ψ_α distribution and constitute a polynomial $e(x)$ of degree $t-1$ such that $e(-i) = e_i$ for $i = 1, 2, \ldots, t$.
(3) Sends $[Q_1(i), \ldots, Q_{sp}(i), e(i)]$ to the i^{th} participant, $1 \leq i \leq n$, as their private share.
(4) Publishes the values $S_i + (\langle A.rot^i[Q_1(0), \ldots, Q_{sp}(0)]\rangle + e(0))\overline{1}_{sp+1}$ for $1 \leq i \leq m$.

Obviously, the sharing process is very simple and this facts makes the scheme applicable and efficient.

4.2. Secret Reconstruction

Now, we explain recovering secret process. Assume t participants $P_{r_1}, P_{r_2}, \ldots, P_{r_t}$ collaborate to recover S_j, hence they compute and submit the related sub-shares using the algorithm $Sub(.)$.

$$Sub([Q_1(r_i), \ldots, Q_{sp}(r_i), e(r_i)], j) = \langle A.rot^j[Q_1(r_i), \ldots, Q_{sp}(r_i)]\rangle + e(r_i).$$

Using these sub-shares and the public values the S_j can be recovered;

$$\sum_{i=1}^{t}\Big((\langle A.rot^j[Q_1(r_i),\ldots,Q_{sp}(r_i)]\rangle + e(r_i))\prod_{k=1,k\neq i}^{t}\frac{r_k}{r_k-r_i}\Big)$$
$$= \langle A.rot^j[\sum_{i=1}^{t}\big(Q_1(r_i)\prod_{k=1,k\neq i}^{t}\frac{r_k}{r_k-r_i}\big),\ldots,\sum_{i=1}^{t}\big(Q_{sp}(r_i)\prod_{k=1,k\neq i}^{t}\frac{r_k}{r_k-r_i}\big)]\rangle$$
$$+\sum_{i=1}^{t}\big(e(r_i)\prod_{k=1,k\neq i}^{t}\frac{r_k}{r_k-r_i}\big) = \langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0)$$

Consequently,

$$\begin{aligned}S_j =&(S_j + (\langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0))\overline{1}_{sp+1})\\ &- \langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0))\overline{1}_{sp+1}\end{aligned}$$

As you can see, any subset of authorized participants can recover any secret in each stage without any constraint on the order of secret recovering.

4.3. Security

In this section we will prove that until LWE has not any polynomial solution, our scheme cannot be broken in polynomial time. First, we should show that all of $e(i)$, $1 \leq i \leq n$, have ψ distribution.

LEMMA 4.1. *Each of* $e(i)$, $1 \leq i \leq n$, *has* $\psi_{c\alpha}$ *distribution, where* c *is a constant.*

PROOF. For $i \in \{1,2,\ldots,n\}$, $e(i)$ can be written as linear combination of variables that have ψ_α distribution. In other words we have,

$$e(i) = \sum_{r=t}^{t}\big(e(-r)\prod_{s=1,s\neq r}^{t}\frac{i-s}{r-s}\big)$$

Thus, $e(i)$ is linear combination of variables $e(-r)$, $r = 1,2,\ldots,t$, that have distribution ψ_α. Consequently, variable $e(i)$ has $\psi_{c\alpha}$ distribution. □

We will show that if there exists an adversary which can break the presented scheme (see Def. 3.2), then we are able to solve the problem 1.

THEOREM 4.2. *If there exist an adversary* $\mathcal{A}$ *that can break the MSS scheme with non-negligible probability, then we can build an adversary which solves the LWE problem with non-negligible probability.*

PROOF. First, we make an algorithm $\mathcal{B}$ that on input $(A, As + e)$ outputs $(Arot^j(s) + e, j)$ with a non-negligible probability. If we build this algorithm, then due to theorem 2.3 solving the problem LWE would be easy.

Suppose the value $(A, As + e)$ is given. The algorithm $\mathcal{B}$ is built as follows,

ALGORITHM 1: How to write algorithms

Data: $(A, As + e, \mathbb{Z}_q^{(sp+1)})$
Result: $(Arot^j(s) + e, j)$ for a value j
Choose $t, n \in_r \mathbb{N}$ $(t \leq n)$;
Choose an natural number $2 \leq r \leq \log sp$;
Choose a matrix $M_{n\times r}$ with random elements in $\mathbb{Z}_2$;
if
(*1*) *M has at most $t-1$ non-zero rows $i_1, \ldots, i_{t-1}$.*
(*2*) *M has a row such that there exists just one non-zero element in it (suppose the mentioned element is M_{i_k,j_k}).*
then
  for $row = 1$ *to* n **do**
    if $row \neq i_k$ **then**
      Choose $S_{row} \in_r \mathbb{Z}_q^{sp}$;
      Choose $e_{row} \in \mathbb{Z}_q$ according ψ_α distribution;
      /*choosing (S_{row}, e_{row}) as the row^{th} participant's share*/
    end
  end
  for $row = 1$ *to* n **do**
    if $row \neq i_k$ **then**
      for $col = 1$ *to* r **do**
        if $M_{row,col} = 1$ **then**
          feed $Arot^{col}(S_{row}) + e_{row}$ to $\mathcal{A}$ /*As the revealed sub-share*/
        end
      end
    else
      feed $As + e$ to $\mathcal{A}$;
      /*As the revealed sub-share corresponding to element M_{i_k,j_k}*/
    end
  end
  Choose $Pv_1, \ldots, Pv_r \in_r \mathbb{Z}_q^{(sp+1)}$ and feed them to $\mathcal{A}$ /*As public values*/;
  run $\mathcal{A}$ to output S_j;
  if $j \neq j_k$ **then**
    Using $S_j - Pv_j$ and $t-1$ sub-shares $Arot^j(S_{i_c}) + e_{i_c}$, $c \in \{1, 2, \ldots, t\} \setminus \{k\}$ compute SUB, sub-share of P_{i_k} corresponding to S_j, by interpolation. /*$\{i_1, \ldots, i_t\}$ are non-zero rows.*/;
    if $j > j_k$ **then**
      RETURN $(SUB, j - j_k) = (Arot^{j-j_k}(s) + e, j - j_k)$
    end
    if $j < j_k$ **then**
      RETURN $(SUB, sp + j - j_k) = (Arot^{sp+j-j_k}(s) + e, sp + j - j_k)$
    end
  end
  failure;
else
  failure;
end

Before computing success probability, let's explain how this algorithm works. we have an algorithm $\mathcal{A}$ which can break the scheme. Our goal is to build an algorithm $\mathcal{B}$ that on input $(A, As + e)$ outputs $(Arot^j(s) + e, j)$. It was shown that if such an algorithm exists then, we can solve the LWE problem (see theorem (2.3)). The core of the algorithm $\mathcal{B}$ is simulating an appropriate experiment for algorithm $\mathcal{A}$ and achieve the correct answer. In this simulation the role of matrix M is to determine the revealed sub-shares, that is, if $M_{i,j} = 1$ then we feed the sub-share of participant P_i corresponding to the secret S_j to $\mathcal{A}$.

Let's back to computing success probability. The algorithm $\mathcal{B}$ will success if the following conditions happen;

(1) The chosen matrix $M_{n\times r}$ has two mentioned conditions.
(2) The algorithm $\mathcal{A}$ outputs the correct answer.
(3) $j \neq j_k$.

For the first one the probability of choosing a appropriate matrix is at least:

$$\frac{\binom{n}{1}r + \binom{n-1}{t-2}2^{r(t-2)}}{2^{rn}}$$

which is a non-negligible function of sp because,

$$\frac{\binom{n}{1}r + \binom{n-1}{t-2}2^{r(t-2)}}{2^{rn}} \geq \frac{2^{r(t-2)}}{2^{rn}}$$

In addition, $r \leq m \leq \log sp$ thus, $\frac{2^{r(t-2)}}{2^{rn}}$ is non-negligible and t, n are fix, therefore the above probability is non-negligible. For the second one, $\mathcal{A}$ outputs the right answer with non-negligible function $\epsilon(sp)$ and finally, the last item occur with non-negligible probability $\frac{r-1}{r}$. Consequently, the success probability is,

$$\frac{(r-1)\epsilon(sp)(\binom{n}{1}r + \binom{n-1}{t-2}2^{r(t-2)})}{r2^{rn}}$$

Clearly, this probability is non-negligible function of sp. □

Combining this results and the previous ones, we can conclude that until the problem LWE has not a polynomial solution, in ours schemes recovering unrecovered secrets is impossible in polynomial time. In other words, if any algorithm breaks the presented MSS scheme with security parameter sp, then it can solve the LWE problem with the parameter $n = sp$ and if we be able to solve the problem LWE with the parameter $n = sp$ then we can solve the problems $SIVP_{\tilde{O}(sp/\alpha)}$ and $SVP_{\tilde{O}(sp/\alpha)}$ $(n = sp)$. These fact imply that breaking the presented scheme is harder than solving these problems.

5. VERIFIABLE VERSION

This section deal with introducing verifiable version of presented scheme. One of the must important issues in secret sharing scheme is verifiability [Chor et al. 1985]. We can not always trust to the participants because they can submit wrong sub-share thus a method should be presented in which cheaters participant can be identified. The verifiable version is a modified version of what is presented before;

(1) Suppose m secrets are shared as in presented scheme with the following extra conditions.
    (a) p is prime number such that discrete logarithm in $\mathbb{Z}_p^*$ is believed to be hard.

(b) $Q_1(x), Q_2(x), \ldots, Q_{sp}(x), e(x)$ are chosen in such a way that the shares belong to $(\mathbb{Z}_p^*)^{sp+1}$.

(2) Dealer chooses an element $g \neq 1$ form $\mathbb{Z}_p^*$ and publishes the following values along with what is published in sharing secrets process.
(a) $[g^{Q_1(i)}, \ldots, g^{Q_{sp}(i)}, g^{e(i)}]$, $1 \leq i \leq n$.
(b) g^{S_i}, $1 \leq i \leq m$.

Above information able participants to check their shares. In addition, it can be checked that participants submit correct sub-share in recovering stage.

Checking the shares. The i^{th} participant checks if his/her share is the discrete logarithm of $[g^{Q_1(i)}, \ldots, g^{Q_{sp}(i)}, g^{e(i)}]$ or not. Furthermore, they can recognize cheating of the dealer because if the dealer be honest then the following relation would be correct for any $j \in \{1, 2, \ldots, m\}$ and $\{r_1, r_2, \ldots, r_t\} \subset \{1, 2, \ldots, n\}$;

$$g^{S_j} = \frac{g^{(S_j + (\langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0))\overline{1}_{sp+1})}}{g^{\langle A.rot^j[Q_1(0),\ldots,Q_m(0)]\rangle + e(0))\overline{1}_{sp+1}}} \tag{1}$$

Plus, $g^{(\langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0))}$ can computed as follows;

$$\prod_{i=1}^{t} \left(g^{\langle A.rot^j[Q_1(r_i),\ldots,Q_{sp}(r_i)]\rangle + e(r_i)} \right)^{\prod_{k=1,k\neq i}^{t} \frac{r_k}{r_k - r_i}} =$$

$$\prod_{i=1}^{t} \left(g^{\prod_{k=1,k\neq i}^{t} \frac{r_k}{r_k - r_i} \langle A.rot^j[Q_1(r_i),\ldots,Q_{sp}(r_i)]\rangle + e(r_i)} \right) =$$

$$g^{\sum_{i=1}^{t} \left(\prod_{k=1,k\neq i}^{t} \frac{r_k}{r_k - r_i} (\langle A.rot^j[Q_1(r_i),\ldots,Q_{sp}(r_i)]\rangle + e(r_i)) \right)} =$$

$$g^{(\langle A.rot^j[Q_1(0),\ldots,Q_{sp}(0)]\rangle + e(0))}$$

Hence, if the relations 1 are true, then it yields that dealer correctly allocated the shares to the participants. In other word, cheating of the dealer can be determined.

Verifing the submitted sub-shares. In secret sharing scheme dishonest participants may submit wrong sub-share. In presented scheme it can be verify that participants have submitted the correct sub-shares. Suppose in recovering stage of secret S_j the participant P_r has submitted the value v. If v satisfies in the following relation, then it is correct.

$$g^v = \prod_{i=1}^{sp} g^{a_i Q_{i+j(mod\ sp-1)}(r)} g^{e(r)}. \tag{2}$$

Because the correct value of sub-share corresponding to S_j of participant P_r is $\langle A.rot^j[Q_1(r), \ldots, Q_{sp}(r)]\rangle + e(r)$. If $v = \langle A.rot^j[Q_1(r), \ldots, Q_{sp}(r)]\rangle + e(r)$ then we have;

$$g^v = g^{\langle A.rot^j[Q_1(r),\ldots,Q_{sp}(r)]\rangle + e(r)} = g^{a_1 Q_{j+1}(r) + \ldots + a_{sp} Q_1(r)} . g^{e(r)} = \prod_{i=1}^{sp} g^{a_i Q_{i+j(mod\ sp-1)}(r)} g^{e(r)}.$$

We know that discrete logarithm chosen kind of group is infeasible [Stinson 2005]. So, revealing the values in verifiable version do not compromise the security of scheme.

Now, we wrap-up what is said in previous lines. In this section we add the verifiability property to our scheme. This property helps us to determine dishonest participants and the dealer.

6. CONCLUSION

The infrastructural of any cryptographic concept is exact definition of security. Lake of security definition for MSS schemes encouraged us to give an exact definition. Therefore, we give a definition and present an MSS scheme based on well studied LWE problem which satisfies in the presented definition. In the presented scheme, we have no constraint in recovering secrets order, in other word they can reconstruct any secret in any stage. This scheme benefit from assigning just one share to each participant, this makes the scheme more applicable because managing the small shares is easier. For the end of this paper, the verifiable version of the MSS scheme is presented. Verifiability is a highly desirable feature in secret sharing because it prevents cheating.

ACKNOWLEDGMENTS

We would like to express our very great appreciation to Mohammad Ghanoonibagha for his valuable and constructive suggestions during the planning and development of this research work.

REFERENCES

Shweta Agrawal, Dan Boneh, and Xavier Boyen. 2010. Efficient lattice (H) IBE in the standard model. In *Advances in Cryptology-EUROCRYPT 2010*. Springer, 553–572.

Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. 2009. Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography*. Springer, 474–495.

George Robert Blakley. 1899. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*. IEEE Computer Society, 313–313.

Ting-Yi Chang, Min-Shiang Hwang, and Wei-Pang Yang. 2005. A new multi-stage secret sharing scheme using one-way function. *ACM SIGOPS Operating Systems Review* 39, 1 (2005), 48–55.

Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. 1985. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 383–395.

Rachid El Bansarkhani and Mohammed Meziani. 2012. An efficient lattice-based secret sharing construction. In *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*. Springer, 160–168.

L Harn. 1995a. Comment on" Multistage secret sharing based on one-way function". *Electronics Letters* 31, 4 (1995), 262.

Lein Harn. 1995b. Efficient sharing (broadcasting) of multiple secrets. *IEE Proceedings-Computers and Digital Techniques* 142, 3 (1995), 237–240.

Jingrui He and Ed Dawson. 1995a. Multisecret-sharing scheme based on one-way function. *Electronics Letters* 31, 2 (1995), 93–95.

Jingrui He and Ed Dawson. 1995b. Multisecret-sharing scheme based on one-way function. *Electronics Letters* 31, 2 (1995), 93–95.

Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. 2007. Multi-bit cryptosystems based on lattice problems. In *Public Key Cryptography-PKC 2007*. Springer, 315–329.

Daniele Micciancio and Shafi Goldwasser. 2002. *Complexity of lattice problems: a cryptographic perspective*. Vol. 671. Springer.

Daniele Micciancio and Oded Regev. 2009. Lattice-based cryptography. In *Post-quantum cryptography*. Springer, 147–191.

Oded Regev. 2006. Lattice-based cryptography. In *Advances in Cryptology-CRYPTO 2006*. Springer, 131–141.

Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

Markus Stadler. 1996. Publicly verifiable secret sharing. In *Advances in CryptologyEUROCRYPT96*. Springer, 190–199.

Ron Steinfeld, Huaxiong Wang, and Josef Pieprzyk. 2004. Lattice-based threshold-changeability for standard Shamir secret-sharing schemes. In *Advances in Cryptology-ASIACRYPT 2004*. Springer, 170–186.

Douglas R Stinson. 2005. *Cryptography: theory and practice*. CRC press.

Andrew C Yao. 1982. Protocols for secure computations. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 160–164.