Security Analysis of a Dynamic Threshold Secret Sharing Scheme Using Linear Subspace Method

Sadegh Jamshidpour, Zahra Ahmadian

Abstract—A dealer-free and non-interactive dynamic threshold secret sharing scheme has been proposed by Harn et.al., in 2015. In this scheme, a (t, n) secret sharing scheme in secret reconstruction phase can turn into a (m, n) scheme in secret reconstruction phase, where m is the number of participanting shareholders. It has been claimed that the secrecy of shares and the secrecy of the secret are unconditionally preserved if $m \in (t, 1 + t(t + 1)/2]$.
This paper provides a security analysis of this scheme in two directions. Firstly, we show that this scheme does not have the dynamic property, i.e. any $t + 1$ released values are sufficient to reconstruct the secret, even the agreed updated threshold is larger. Secondly, we show that any $t + 1$ released values are sufficient to forge the released value of a non-participating shareholder.
The technique that we enjoyed for our analysis is the linear subspace method, which basically measures the information leaked by the known parameters of the scheme by computing the dimension of the linear subspace spanned by these parameter. This method has shown to be capable of cryptanalysis of some secret sharing based schemes, whose security relies on keeping the coefficients of the underlying polynomial(s) secret.

Index Terms—Dynamic threshold, Linear subspace, Forging, Secret reconstruction.

1 Introduction

Shamir [1] and Blakley [2] independently proposed the first threshold secret sharing schemes in 1979. In a threshold secret sharing scheme the secret is divided into n shares such that: (1) any t or more than t shares can recover the secret, and (2) any less than t shares gives no information from the secret.

The idea of changing the threshold parameter of a secret sharing scheme, i.e. converting a (t,n)-threshold scheme into a (t',n)-threshold scheme, where $t < t' \leq n$, was first proposed by Martin et al. in 1999 [3]. In such a scheme the threshold of the scheme can be increased to t', while shares have been previously distributed based on the old threshold t. Such schemes are called *threshold changeable* or *dynamic threshold* secret sharing schemes.

Changing the threshold may be required for protecting the secret when the security policy of the scheme changes after distributing shares. This may be caused by joining or leaving participants, change in the mutual trust among participants over time, etc. [3], [4]. In a different but very concrete approach, this direction is followed focusing on the communication efficiency or the so-called decoding bandwidth in [5], [6]. In more details, the excess of participating shareholders is regarded as an opportunity to minimize the amount of communications between the shareholders, i.e. minimizing the size of values released by shareholders in the secret reconstruction phase.

There is another advantage behind the possibility of changing the threshold of a secret sharing scheme. A well-known attack on the Shamir's secret sharing scheme is as follows. Suppose that there are more than t shareholders participating in the secret reconstruction phase. Since the threshold of the secret is t, a non-shareholder who impersonates to be a valid shareholder, can forge the share of the victim shareholder and even gain the secret by gathering t shares from other participating shareholders. Besides the user authentication or Verifiable Secret Sharing (VSS) techniques [7], [8], one approach to prevent this security threat could be increasing the threshold of the scheme exactly up to the number of alleged participating shareholders, including the possible non-shareholder(s) [9]. In this way, the maximum number of shares that the attacker can obtain would be strictly less than the new threshold which is not sufficient for the non-shareholder to mount the above attack.

Regardless of the applications and advantages, it is clear that in the new situation (i.e. when the threshold have been changed into $t' > t$) the values released by the shareholders in the secret reconstruction phase, are not the shares themselves, but they are actually functions of them which we call *tokens*. Additionally, the security definition of the scheme is updated: (1) any t' or more than t' tokens can completely recover the secret and (2) any less than t' tokens should not give any information from the secret.

In this direction, Harn et al. [9] proposed a dealer-free and non-interactive dynamic threshold secret sharing scheme in which the secret can only be reconstructed when all participating shareholders actively act in the secret reconstruction. More precisely, it has been claimed that the

- *Sadegh Jamshidpour is with the Department of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran. Email: s.jamshidpour@stu.nit.ac.ir*
- *Zahra Ahmadian is with the Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran. E-mail: z_ahmadian@sbu.ac.ir*

threshold of the secret reconstruction changes into m, the number of participants, while shares are previously distributed using a smaller threshold t. The scheme utilizes a symmetric bivariate polynomial of degree $t-1$ with $t(t+1)/2$ coefficients for the share generation, and the secret is reconstructed using Lagrange interpolation formula. To preserve the secrecy of the secret and the secrecy of shares, the number of participants is limited to be in the interval $(t, t(t+1)/2+1]$.

The security of this dynamic threshold secret sharing scheme [9] relies on keeping the $t(t+1)/2$ coefficients of the underlying polynominal confidential. This approach can also be found in some work including a group authentication protocol [10], a secret sharing scheme resistant against active adversaries [11], and a group authentication for LTE network [12].

This approach has been challenged in a recent work [13] by proposing a new cryptanalysis method tailored to such structures, called the *linear subspace cryptanalysis*. In [13], it has been shown that an impersonation attack can be successfully mounted on the secret sharing-based group authentication protocol proposed in [10]. The linear subspace cryptanalysis shows that assuring that the polynomials coefficients can not be retrieved, does not guarantee that the protocol is secure.

In the same vein, we provide a security analysis of the dynamic threshold secret sharing scheme proposed in [9] with the aim of challenging its security claims. We first provide an algebraic representation of the target scheme which not only clarifies the structure of the scheme more precisely, but also is a good tool for evaluation of the amount of information leaked from the scheme. Then, the secrecy of the secret and shares, are made equivalent to unsolvability of two (similar) systems of linear equations. The rank study of these systems shows that the dimension of the subspace spanned by all the values released by the participating shareholders, which are called tokens, never exceeds $t+1$. Motivated from this fact, we demonstrate that the values released by any $t+1$ users can form a basis for the mentioned subspace and therefore the information of this basis is sufficient to retrieve the secret and all other tokens, without having the unknown coefficients of the symmetric bivariate polynomial. So, it is concluded that the secret sharing scheme proposed in [9] fails to provide the secrecy of the secret and the secrecy of tokens.

The paper is organized as follows. Section II contains preliminaries including Shamir's secret sharing scheme, Harn et al. dynamic threshold secret sharing scheme and the algebraic representation of the dynamic scheme. In Section III, the security analysis given by the designers is discussed in detail and its invalidity is shown. Then, in Section IV, we provide our security analysis of the scheme including the justification that shows how this scheme fails as a dynamic threshold scheme. Finally, section V concludes the paper

2 Dynamic Threshold Secret Sharing Scheme

In this section we first review Shamir's (t,n) secret sharing scheme [1], as the root of the analyzed dynamic threshold secret sharing scheme [9]. Then a generalization of the dynamic threshold secret sharing scheme of [9] and its original variant as a special case are introduced. Finally, we propose our algebraic representation of the generalized scheme which brings us a precise description of the analyzed system. For the both secret sharing schemes introduced in the following, we assume that there are n shareholders, $\mathcal{U} = \{U_1, U_2, ..., U_n\}$, and a mutually trusted dealer who is responsible for generating shares and distributing them among the shareholders, securely.

2.1 Shamir's (t,n) secret sharing scheme

Shamir's (t,n) secret sharing scheme [1] splits a secret into n shares with the property that any t or more than t shares can recover the secret, while any less than t shares reveals no information about the secret. The scheme consists of two phases:

2.1.1 Share generation phase

The dealer first selects a random univariate polynomial, $f(x) = a_0 + a_1 x + ... + a_{t-1}x^{t-1} \bmod p$, of degree $t-1$, with coefficients $a_i \in Z_p, i = 0, ..., t-1$, such that the secret satisfies $s = f(0)$. Then the dealer generates shares $f(x_i)$, $i = 1, ..., n$, where $x_i \in Z_p$ is the public information associated with shareholder U_i. Finally the dealer distributes each share $f(x_i)$ to the corresponding shareholder U_i, in a secure way.

2.1.2 Secret reconstruction phase

Once m shareholders, $\mathcal{U}' = \{U_{i_1}, U_{i_2}, ..., U_{i_m}\}, t \le m \le n$ want to reconstruct the secret, each shareholder releases his share to all other participating shareholders. By knowing of m shares, each shareholder computes the secret using Lagrange interpolation formula as following:

$$s = f(0) = \sum_{i=1}^{m} f(x_i) \prod_{j \in I, j \neq i} \frac{-x_j}{x_i - x_j} \bmod p. \qquad (1)$$

where $I = \{i_1, i_2, \ldots, i_m\}$.

2.2 The generalized dynamic threshold secret sharing scheme

Harn et al. [9] proposed a dynamic threshold secret sharing scheme in which the threshold of the scheme can dynamically change from t which is the threshold set in the share generation phase into the number of participating members $t < m \le 1 + t(t+1)/2$ in the secret reconstruction phase. To facilitate our proposed analysis and generalize its results, we first introduce a generalization of the dynamic threshold secret sharing scheme [9]. Then, the original scheme is presented as a special case. The generalized scheme includes two phases:

2.2.1 Share generation phase

The dealer first selects a symmetric bivariate polynomial with degree $t-1$ as follows:

$$F(x, y) = \sum_{0 \le i,j \le t-1} b_{i,j} x^i y^j \bmod p \qquad (2)$$

where $b_{i,j} \in Z_p$, $b_{i,j} = b_{j,i}$, $\forall i,j \in [0, t-1]$, such that the secret $s \in Z_p$ satisfies:

$$s = F(u_1, v_1) + dF(u_2, v_2) \bmod p \tag{3}$$

where $d, u_1, u_2, v_1, v_2 \in Z_p$, $u_1 \neq u_2$, $v_1 \neq v_2$, are public information. Next the dealer generates shares $s_i(y) = F(x_i, y) \bmod p$, $i = 1, ..., n$, where $x_i \in Z_p$ and $x_i \notin \{u_1, u_2\}$ is the public information associated with shareholder U_i. Finally the dealer distributes each share $s_i(y)$, which is an univariate polynomial of degree $t-1$, to the corresponding shareholder U_i, securely.

2.2.2 Secret reconstruction phase

Once m shareholders $\mathcal{U}' = \{U_{i_1}, U_{i_2}, ..., U_{i_m}\}$, $t < m \leq 1 + t(t+1)/2$ want to reconstruct the secret, each one uses his share $s_i(y)$ and public information d, u_1, u_2, v_1, v_2, announced by the dealer, to calculate a *token* (also called *released value* in [9]) as following:

$$c_i = F(x_i, v_1) \prod_{j \in I, j \neq i} \frac{u_1 - x_j}{x_i - x_j} + dF(x_i, v_2) \prod_{j \in I, j \neq i} \frac{u_2 - x_j}{x_i - x_j} \bmod p, \tag{4}$$

where $I = \{i_1, i_2, \ldots, i_m\}$. Then he issues his token to all other participants. Finally, each participant having m tokens calculates the secret as follows:

$$s = \sum_{i=1}^{m} c_i \bmod p. \tag{5}$$

Note that the assumption of using arbitrary values for u_1, u_2, v_1, v_2 has not been made in the original scheme, where a special choice satisfying the condition $u_1 \neq u_2$, $v_1 \neq v_2$ has been utilized for these variables as $(u_1, v_1) = (0, 0)$ and $(u_2, v_2) = (1, 1)$. We provided this simple generalization to facilitate our next security analysis and make sure that our results do not depend on special choices for public parameters u_1, u_2, v_1, v_2. Throughout this paper, we work on this generalized scheme. Obviously, the results are valid for the original scheme given in [9], as a special case.

2.3 Algebraic representation of the scheme

This section provides a new representation of the dynamic threshold secret sharing scheme, called algebraic representation, which equips us with efficient algebraic tools for a precise security analysis of the scheme. W.l.g., consider that the set of m shareholders participating in the secret reconstruction is $\mathcal{U}' = \{U_1, U_2, ..., U_m\}$, there is a non-shareholder among which gathering the released tokens for his later efforts. Our goal is to find an explicit representation for the information obtained by the non-shareholder.

Let us first rewrite the symmetric bivariate polynomial in (2) as following:

$$F(x, y) = \mathbf{y}^{\mathrm{T}} \mathbf{B} \mathbf{x}, \tag{6}$$

where:

$$\mathbf{x} = \left[1, x, x^2, ..., x^{t-1}\right]^{\mathrm{T}},$$

$$\mathbf{y} = \left[1, y, y^2, ..., y^{t-1}\right]^{\mathrm{T}}.$$

Symmetric Matrix $[\mathbf{B}]_{i,j} = b_{i,j}$, $\forall i,j \in [0, t-1]$, is the matrix of the coefficients of the bivariate polynomial, in which the symmetry property implies that $\mathbf{B}^{\mathrm{T}} = \mathbf{B}$ (the superscript "T" denotes the transpose operator).

Using the new representation of the bivariate polynomial given in (6), each token c_i can be rewritten as following:

$$c_i = \mathbf{v}_1^{\mathrm{T}} \mathbf{B} \mathbf{x}_i l_{1,i} + d \mathbf{v}_2^{\mathrm{T}} \mathbf{B} \mathbf{x}_i l_{2,i} \tag{7}$$

where:

$$\begin{aligned} \mathbf{x}_i &= \left[1, x_i, \ldots, x_i^{t-1}\right]^{\mathrm{T}}, \ i = 1, ..., m, \\ \mathbf{v}_r &= \left[1, v_r, \ldots, v_r^{t-1}\right]^{\mathrm{T}}, \ r = 1, 2, \end{aligned}$$

and

$$l_{r,i} = \prod_{j=1, j \neq i}^{m} \frac{u_r - x_j}{x_i - x_j} \bmod p, \ \ r = 1, 2.$$

The representation of token given in (7) can be simplified using block matrix operations as follows:

$$\begin{aligned} c_i &= \begin{bmatrix} \mathbf{v}_1^{\mathrm{T}} \mathbf{B} & d \mathbf{v}_2^{\mathrm{T}} \mathbf{B} \end{bmatrix} \begin{bmatrix} l_{1,i} \mathbf{x}_i \\ l_{2,i} \mathbf{x}_i \end{bmatrix} \\ &= \mathbf{b}(\mathbf{l}_i \otimes \mathbf{x}_i) \end{aligned} \tag{8}$$

where the symbol " $\otimes$ " denotes the Kronecker product and:

$$\mathbf{b} = \begin{bmatrix} \mathbf{v}_1^{\mathrm{T}} \mathbf{B} & d \mathbf{v}_2^{\mathrm{T}} \mathbf{B} \end{bmatrix}, \tag{9}$$

$$\mathbf{l}_i = \begin{bmatrix} l_{1,i} & l_{2,i} \end{bmatrix}^{\mathrm{T}}. \tag{10}$$

Now, we gather all tokens, c_i, $i = 1, ..., m$, into a vector $\mathbf{c}$, called the tokens vector, as follows:

$$\mathbf{c} = \begin{bmatrix} c_1, & c_2, & \cdots, & c_m \end{bmatrix}. \tag{11}$$

Substituting equation (8) into the tokens vector $\mathbf{c}$ gives:

$$\begin{aligned} \mathbf{c} &= \begin{bmatrix} \mathbf{b}(\mathbf{l}_1 \otimes \mathbf{x}_1), \ \mathbf{b}(\mathbf{l}_2 \otimes \mathbf{x}_2), \ \ldots, \ \mathbf{b}(\mathbf{l}_m \otimes \mathbf{x}_m) \end{bmatrix} \\ &= \mathbf{b} \begin{bmatrix} \mathbf{l}_1 \otimes \mathbf{x}_1, \ \mathbf{l}_2 \otimes \mathbf{x}_2, \ \ldots, \ \mathbf{l}_m \otimes \mathbf{x}_m \end{bmatrix} \\ &= \mathbf{b}\mathbf{M} \end{aligned} \tag{12}$$

where:

$$\mathbf{M} = \begin{bmatrix} \mathbf{l}_1 \otimes \mathbf{x}_1, & \mathbf{l}_2 \otimes \mathbf{x}_2, & \cdots, & \mathbf{l}_m \otimes \mathbf{x}_m \end{bmatrix}. \tag{13}$$

Matrix $\mathbf{M}$ can be represented as the Khatri-Rao product [14], denoted by the symbol " $\odot$ ", of matrices $\mathbf{L}$ and $\mathbf{X}$ as following:

$$\mathbf{M} = \mathbf{L} \odot \mathbf{X}. \tag{14}$$

Where $\mathbf{X}$ is a $t \times m$ Vandermonde matrix with elements x_i, $i = 1, ..., m$, as following:

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} \mathbf{x}_1, & \mathbf{x}_2, & ..., & \mathbf{x}_m \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_m \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{t-1} & x_2^{t-1} & \ldots & x_m^{t-1} \end{bmatrix}, \end{aligned} \tag{15}$$

and matrix $\mathbf{L}$ of size $2 \times m$ is as follows:

$$\mathbf{L} = \begin{bmatrix} \mathbf{l}_1, & \mathbf{l}_2, & \cdots, & \mathbf{l}_m \end{bmatrix}. \tag{16}$$

Expression (12) describes the publicly available tokens in terms of a known matrix, $\mathbf{M}$, with an explicit structure, and a vector, $\mathbf{b}$, constructed from unknown coefficients of the bivariate polynomial and public information v_1, v_2 and

d. This description plays a vital role in our security analysis of the scheme.

Finally, the secret s, given in equation (3), can be represented in matrix notation as following:

$$\begin{aligned} s &= \mathbf{v}_1^{\mathrm{T}}\mathbf{B}\mathbf{u}_1 + d\mathbf{v}_2^{\mathrm{T}}\mathbf{B}\mathbf{u}_2 \\ &= \begin{bmatrix} \mathbf{v}_1^{\mathrm{T}}\mathbf{B} & d\mathbf{v}_2^{\mathrm{T}}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \\ &= \mathbf{b}\mathbf{u}, \end{aligned} \tag{17}$$

where:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}, \tag{18}$$

and $\mathbf{u}_r = \left[1, u_r, ..., {u_r}^{t-1}\right]^{\mathrm{T}}$, $r = 1, 2$.

This algebraic representation would be useful for evaluating the information leaked from the released tokens. Next section concerns the analysis of this information and explores the incorrectness of the security analysis provided in [9].

3 Evaluation of Security Analysis of Designers

The dynamic threshold secret sharing scheme [9] has been designed with the aim of protecting the secrecy of the secret and the secrecy of shares from non-shareholders, while there are $m > t$ participants in the secret reconstruction. These goals have been assumed to be reached by protecting the coefficients of the bivariate polynomial by imposing the bound $m \in (t, 1 + t(t+1)/2]$ on the number of participants.

According to the security analysis in [9], each valid token gives a linear function in $t(t+1)/2$ polynomial coefficients $b_{i,j}, 0 \leq i, j \leq t-1$. That is also what (12) says. It is expressed in [9] that with such a bound for m, the non-shareholder can gather at most $t(t+1)/2 - 1$ valid tokens. Thus, all he can obtain is a system of linear equations, in which the number of unknowns is more than the number of equations. So, it is impossible to uniquely derive the bivariate polynomial coefficients, and since the secret and shares were generated by the polynomial, the secrecy of the secret and shares are preserved *unconditionally*, as claimed in [9]. In other words, the secrecy of the secret and the secrecy of shares have been assumed to be equivalent to the secrecy of the bivariate polynomial.

Before treating this approach more precisely, we have a primary discussion about the bound given in [9] for m. The fact is that, according to the rational of [9], the correct upper bound for m must be $t(t+1)/2$ not $1 + t(t+1)/2$. In order for the non-shareholder to gather strictly less than $t(t+1)/2$ equations, the number of participants (including at least one non-shareolder) must be at most $t(t+1)/2$. In this way, the number of valid released tokens would be at most $t(t+1)/2 - 1$ which certainly makes the system of equations in (12) underdetermined.

However, to analyze the solvability of (12) in a more precise approach, what should be discussed first is the rank of matrix $\mathbf{M}$, which we prove in the rest of this section. We first present the Rank-Nullity theorem as an important tool in our analysis. Then, the rank analysis is provided in Theorem 2.

Theorem 1 (Rank-Nullity theorem). Let $\mathbf{A}$ be a $m \times n$ matrix. Then,

$$dim(C(\mathbf{A})) + dim(N(\mathbf{A})) = n. \tag{19}$$

where $C(\mathbf{A})$ and $N(\mathbf{A})$ represent the column space and nullspace of matrix $\mathbf{A}$, respectively.

Proof: A proof can be found in [15]. □

Theorem 2. Let $\mathbf{M} = \mathbf{L} \odot \mathbf{X}$, where $\mathbf{L}$ and $\mathbf{X}$ are defined in (16) and (15), respectively. Then, $rank(\mathbf{M}) = t + 1$ for all $m \geq t + 1$.

Proof: First of all, we perform some row and column operations to simplify matrix $\mathbf{M}$, then the rank of the simplified matrix would be calculated. The row and column operations applied on $\mathbf{M}$ are as follows:

1) multiply column i, where $i = 1, ..., m$, of matrix $\mathbf{M}$ by $(u_1 - x_i)(u_2 - x_i) \prod_{j=1, j\neq i}^{m} (x_i - x_j) \bmod p$.
2) multiply the first t rows by $(\prod_{j=1}^{m} (u_1 - x_j))^{-1} \bmod p$
3) multiply the second t rows by $(\prod_{j=1}^{m} (u_2 - x_j))^{-1} \bmod p$.

After these operations, the resulting matrix $\mathbf{M}^{(1)}$ would be as follows.

$$\mathbf{M}^{(1)} = \begin{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ x_1^{t-1} \end{bmatrix} & \cdots & \begin{bmatrix} 1 \\ \vdots \\ x_m^{t-1} \end{bmatrix} \\ (u_2 - x_1) & & & (u_2 - x_m) \\ & \begin{bmatrix} 1 \\ \vdots \\ x_1^{t-1} \end{bmatrix} & \cdots & \begin{bmatrix} 1 \\ \vdots \\ x_m^{t-1} \end{bmatrix} \\ (u_1 - x_1) & & & (u_1 - x_m) \end{bmatrix}. \tag{20}$$

According to Theorem 1, in order to obtain the rank or equivalently dimension of the column space of matrix $\mathbf{M}^{(1)}$, it is sufficient to find the dimension of its left-nullspace. For this purpose, assume the system:

$$\boldsymbol{\eta}\mathbf{M}^{(1)} = \mathbf{0} \tag{21}$$

where $\eta = \left[\eta_0, \eta_1, ..., \eta_{t-1}, {\eta'}_0, {\eta'}_1, ..., {\eta'}_{t-1}\right]$ is a non-zero vector. It can be simply shown that system (21) is equivalent to:

$$p(x) = 0 \bmod p \ \text{ for } \ x = x_i, \ \ i = 1, ..., m, \tag{22}$$

where $p(x)$ is a t-degree polynomial as following:

$$p(x) = (u_2 - x)\eta(x) + (u_1 - x)\eta'(x) \bmod p, \tag{23}$$

and $\eta(x)$ and $\eta'(x)$ are two $(t-1)$-degree polynomials as follows:

$$\eta(x) = \eta_0 + \eta_1 x + ... + \eta_{t-1}x^{t-1} \bmod p, \tag{24}$$

$$\eta'(x) = \eta'_0 + \eta'_1 x + ... + \eta'_{t-1}x^{t-1} \bmod p. \tag{25}$$

Rearranging $p(x)$ as a standard polynomial gives:

$$p(x) = p_0 + p_1 x + \cdots + p_t x^t \bmod p \tag{26}$$

where

$$p_i = \begin{cases} u_2\eta_0 + u_1\eta'_0, & i = 0 \\ u_2\eta_i + u_1\eta'_i - \eta_{i-1} - \eta'_{i-1}, & 1 \leq i \leq t-1 \\ -\eta_{t-1} - \eta'_{t-1}, & i = t \end{cases} \tag{27}$$

As we have assumed, $m \geq t+1$. For polynomial $p(x)$ of degree t to have $m \geq t+1$ roots, it is necessary that all its $t+1$ coefficients are zero, otherwise the condition (22) is never satisfied. If we force all the coefficients $p_0, \ldots, p_t$ given in (27) to be zero, the $t+1$ resulting linear equations can be expressed in a matrix notation as follows.

$$\mathbf{U}\boldsymbol{\eta} = \mathbf{0} \tag{28}$$

where $\mathbf{U}$ is a $(t+1) \times 2t$ matrix as follows:

$$\mathbf{U} = \begin{bmatrix} u_2 & 0 & \cdots & 0 & u_1 & 0 & \cdots & 0 \\ -1 & u_2 & \cdots & 0 & -1 & u_1 & \cdots & 0 \\ 0 & -1 & \cdots & 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_2 & 0 & 0 & \cdots & u_1 \\ 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & -1 \end{bmatrix}. \tag{29}$$

The main implication is that a vector $\boldsymbol{\eta}$ solves system (28) if and only if it is a solution for system (21). In other words, the nullspace of matrix $\mathbf{U}$ and the left-nullspace of matrix $\mathbf{M}^{(1)}$ are the same.

$$N(\mathbf{M}^{(1)^{\mathrm{T}}}) = N(\mathbf{U}). \tag{30}$$

Consequently,

$$dim(N(\mathbf{M}^{(1)^{\mathrm{T}}})) = dim(N(\mathbf{U})). \tag{31}$$

Therefore the problem of finding the left-nullspace of matrix $\mathbf{M}^{(1)}$ is simplified to finding the nullspace of matrix $\mathbf{U}$ which has a more simple structure. The following Lemma computes the dimesion of the column space of matrix $\mathbf{U}$ which leads us to the dimension of its nullspace, using Theorem 1.

Lemma 1. Matrix $\mathbf{U}$, defined in (29), is full column rank for every $u_1 \neq u_2$.

Proof: Consider matrix $\mathbf{E} = \mathbf{U}(:, [1 : t, 2t])$ constructed from the first t columns and the last column of matrix $\mathbf{U}$. It is sufficient to show that matrix $\mathbf{E}$ is full rank. If so, matrix $\mathbf{U}$ would be obviously of full column rank.

We use Gaussian elimination to calculate rank of matrix $\mathbf{E}$. For this purpose, we first perform the following iterative row operation on all rows of $\mathbf{E}$ to arrive at an eliminated matrix called $\mathbf{E}^{(1)}$.

$$\mathbf{e}_{k+1}^{(1)} = (u_2^{-1}\mathbf{e}_k^{(1)} + \mathbf{e}_{k+1}) \mod p, k = 1, ..., t \tag{32}$$

where $\mathbf{e}_k$ and $\mathbf{e}_k^{(1)}$ denote the k^{th} rows of matrices $\mathbf{E}$ and $\mathbf{E}^{(1)}$, respectively, and $\mathbf{e}_1^{(1)} = \mathbf{e}_1$. The resulting matrix $\mathbf{E}^{(1)}$ would be as follows.

$$\mathbf{E}^{(1)} = \begin{bmatrix} u_2 & 0 & \cdots & 0 & 0 \\ 0 & u_2 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_2 & u_1 \\ 0 & 0 & \cdots & 0 & (u_1/u_2) - 1 \end{bmatrix}.$$

If $u_1 \neq u_2$, the column vectors of matrix $\mathbf{E}^{(1)}$ are definitely in different directions, which implies that its rank is full. So, matrices $\mathbf{E}$ and $\mathbf{U}$ are consequently of full column rank, as well. Thus, the proof is completed. □

Now we use the Rank-Nullity theorem and the result of Lemma 1 to compute the dimension of the nullspace of matrix $\mathbf{U}$ as following:

$$dim(N(\mathbf{U})) = 2t - (t+1) = t - 1.$$

Based on equality (31), we find that:

$$dim(N(\mathbf{M}^{(1)^{\mathrm{T}}})) = dim(N(\mathbf{U})) = t - 1.$$

Finally, applying the Rank-Nullity theorem for matrix $\mathbf{M}^{(1)^{\mathrm{T}}}$ provides:

$$rank(\mathbf{M}) = rank(\mathbf{M}^{(1)}) = dim(C(\mathbf{M}^{(1)^{\mathrm{T}}})) = 2t-(t-1) = t+1.$$

□

Therefore the system of linear equations given in (12) has $t(t+1)/2$ unknowns, while the rank of coefficient matrix is $t+1$ for $m > t$ and t for $m = t$. This means that this system of equations is always underdetermined for any arbitrary m. Hence, despite the strategy taken in [9], there is no need to put an upper bound on m to make sure that the unknown polynomial coefficients can not be recovered uniquely. In fact, for any $m > t$ this system has $p^{t(t+1)/2-(t+1)}$ set of solutions, one of which is the correct set of values for the polynomial coefficients.

But, the main discussion of this paper is that keeping the polynomial coefficients secret does not merely make the scheme secure, necessarily. This claim would be discussed in details in the next section, where we show the incorrectness of the assumption in [9] by demonstrating that how a non-shareholder can break the secrecy of the secret and/or shares, while the above rank analysis of $\mathbf{M}$ guaranties the protection of the coefficients of the bivariate polynomial.

4 Security Analysis of the Scheme

Although Theorem 2 shows that the coefficients of the polynomial can never be retrieved uniquely, its result can still be used for cryptanalytic purposes. First suppose a $2t$-dimensional vector space whose basis is the (unknown) elements of $\mathbf{b}$. We are interested in the subspace spanned by the available tokens as the total information obtained from the legitimate shareholders. The dimension of this subspace is determined by the rank of matrix $\mathbf{M}$, according to equation (12). So, Based on the result of Theorem 2, as long as $m \geq t+1$, the dimension of the subspace spanned by tokens would be always equal to $t+1$ which is independent of m. In other words, having more than $t+1$ valid tokens provides no additional information from those that obtained from $t+1$ ones of them.

So, by having $t+1$ valid tokens, one can bulid a basis, i.e. the corresponding columns of matrix $\mathbf{M}$, for the underlying $t+1$ dimensional subspace. This observation leads us to two results:

1) Any other token must be presented as a linear combination of this set of $t+1$ basis (tokens).
2) The secret itself, as a linear combination of m tokens (5), must be presented as a linear combination of this set of $t+1$ basis (tokens).

In the rest of this section we present a systematic approach to show that how a non-shareholder who has no

valid shares can recover the secret or forge an arbitrary token, just by having a number of $t+1$ valid tokens released by legitimate shareholders. For this purpose, he should find the appropriate linear combinations of the basis to recover the secret and forge the token. Let us first analyze the secrecy of the secret.

4.1 The secrecy of the secret

Suppose that there are m legitimate shareholders, $\mathcal{U}' = \{U_1, U_2, ..., U_m\}$, and a non-shareholder who just listens to the released values, c_i, $i = 1, ..., m$, and aims to recover the secret. For this purpose, assume that there is a vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_m]^\mathsf{T}$ that solves the following system:

$$\mathbf{M}\boldsymbol{\alpha} = \mathbf{u} \tag{33}$$

where matrix $\mathbf{M}$ and vector $\mathbf{u}$ have been defined in (13) and (18), respectively. Thus, one can simply show that multiplying equation (12) by the solution vector $\boldsymbol{\alpha}$ from right side yields.

$$\begin{aligned} \mathbf{c}\boldsymbol{\alpha} &= \mathbf{b}\mathbf{M}\boldsymbol{\alpha} \\ &= \mathbf{b}\mathbf{u} \\ &= s, \end{aligned}$$

which means that if such an α exists, the secret can be described as a linear combination of the tokens. Thus, if there exists a solution for system (33), a non-shareholder can successfully recover the secret by computing a linear combination of the released tokens without retrieving the unknown coefficients of the bivariate polynomial, directly. The following theorem provides the solvability analysis of system (33).

Theorem 3. The system of linear equations $\mathbf{M}\boldsymbol{\alpha} = \mathbf{u}$, is always solvable for every $m \geq t+1$, where matrix $\mathbf{M}$ and vector $\mathbf{u}$ are defined according to (13) and (18), respectively.

Proof: The system is solvable if vector $\mathbf{u}$ belongs to the column space of matrix $\mathbf{M}$. In order to investigate if $\mathbf{u} \in C(\mathbf{M})$, we construct the augmented matrix $\mathbf{R} = [\mathbf{M} \mid \mathbf{u}]$ of size $2t \times (m+1)$, and check if

$$rank(\mathbf{R}) = rank(\mathbf{M}) = t+1 \quad \text{for } m \geq t+1.$$

In other words, by adding vector $\mathbf{u}$ to matrix $\mathbf{M}$ its rank does not increase. In order to compute rank of matrix $\mathbf{R}$, we utilize a similar approach taken in Theorem 2.

We first do the same row and column operations on $\mathbf{R}$ as in Theorem 2. Note that the first column operation is performed on the first m columns of $\mathbf{R}$ and not its last column. The resulting matrix would be as following.

$$\mathbf{R}^{(1)} = \left[\begin{array}{c|cc} & \prod_{j=1}^{m}(u_1 - x_j)^{-1} & \begin{bmatrix} 1 \\ u_1 \\ \vdots \\ u_1^{t-1} \end{bmatrix} \\ \mathbf{M}^{(1)} & & \\ & \prod_{j=1}^{m}(u_2 - x_j)^{-1} & \begin{bmatrix} 1 \\ u_2 \\ \vdots \\ u_2^{t-1} \end{bmatrix} \end{array}\right]$$

To compute rank of matrix $\mathbf{R}^{(1)}$, we explore for the dimension of its left-nullspace. Let us construct system:

$$\boldsymbol{\eta}\mathbf{R}^{(1)} = \mathbf{0}, \tag{34}$$

where $\boldsymbol{\eta} = [\eta_0, \eta_1, ..., \eta_{t-1}, \eta'_0, \eta'_1, ..., \eta'_{t-1}]$ is a non-zero vector. This system is equivalent to the both following conditions.

$$p(x) = 0 \mod p, \quad x = x_i, \ i = 1, ..., m, \tag{35a}$$

$$\eta(u_1)\prod_{j=1}^{m}(u_1 - x_j)^{-1} + \eta'(u_2)\prod_{j=1}^{m}(u_2 - x_j)^{-1} = 0, \tag{35b}$$

where $p(x)$, $\eta(x)$ and $\eta'(x)$ are the same as those defined in (22), (23) and (24), respectively. We intend to show that any solution of (35a) also solves (35b). Thus, it is sufficient to find the solutions of (35a) only. For condition (35a), if we consider the case $m \geq t+1$, polynomial $p(x)$ of degree t can not have m zeros, unless its coefficients are zero all. In this case, it is obvious that its evaluation at any point such as u_1 and u_2 must be zero, as well. So,

$$p(u_1) = p(u_2) = 0. \tag{36}$$

Moreover, it can be simply seen from (22) that evaluation of polynomial $p(x)$ in points u_1 and u_2 are as following.

$$p(u_1) = (u_2 - u_1)\eta(u_1) \tag{37a}$$

$$p(u_2) = (u_1 - u_2)\eta'(u_2). \tag{37b}$$

Since $u_1 \neq u_2$, condition (36) along with (37a) and (37b) implies that:

$$\eta(u_1) = \eta'(u_2) = 0 \tag{38}$$

Thus (35b) is satisfied.

Therefore, every vector $\boldsymbol{\eta}$ that satisfies condition (35a) also fullfills (35b). In other words, it is just required to find solutions of equation (35a), which is also equivalent to problem (22). Thus both matrices $\mathbf{R}^{(1)}$ and $\mathbf{M}^{(1)}$ have the same left-nullspace, which implies that they have the same rank. Finally, we conclude that:

$$rank(\mathbf{R}) = rank(\mathbf{R}^{(1)}) = t+1 \quad \text{for } m \geq t+1.$$

□

Based on the result of theorem 3, the system (33) is always solvable for every $m \geq t+1$. Therefore, a non-shareholder can gahter $t+1$ released tokens and construct the system of linear equations (33) and explore for a solution in a polynomial time, using by e.g. Gaussian elimination method. Then, combining the released tokens according to the solution vector $\boldsymbol{\alpha}$ yields the secret. It is worth to remark that in the process of secret reconstruction, there is no need to find the coefficients of the bivariate polynomial, instead, we look for an appropriate combination of the released tokens to gain the secret.

4.2 The secrecy of shares

In this section we show that how an attacker can forge the released token of an arbitrary victim shareholder without retrieving his share. So, we actually analyze the secrecy of tokens rather than the secrecy of shares, since a non-shareholder only needs to forge a valid token to impersonate

a shareholder, even he does not know the share of victim non-shareholder.

Assume that there are m participants, where $m-1$ of them, $\mathcal{U}'' = \{U_1, U_2, ..., U_{m-1}\}$, are shareholders, and there is a non-shareholder who aims to impersonate the shareholder U_m by forging a token at point x_m. For this purpose, he announces the victim shareholder's identity x_m and waits to receive tokens released by legitimate shareholders. The received tokens can be written as following.

$$\mathbf{c}(1:m-1) = \mathbf{b}\mathbf{M}(:, 1:m-1) \tag{39}$$

where $\mathbf{M}$ and vectors $\mathbf{c}$ and $\mathbf{b}$ are as defined according to (13), (11) and (9), respectively and $\mathbf{c}(1:m-1)$ is the first $m-1$ element of $\mathbf{c}$ and $\mathbf{M}(:, 1:m-1)$ is a matrix composed of the first $m-1$ columns of $\mathbf{M}$. The non-shareholder must produce a valid token at point x_m as follows.

$$\begin{aligned} c_m &= \mathbf{b}(\mathbf{l}_m \otimes \mathbf{x}_m) \\ &= \mathbf{b}\mathbf{M}(:, m). \end{aligned} \tag{40}$$

where $\mathbf{M}(:, m)$ is the last column of $\mathbf{M}$. Now, assume that there is vector $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_{m-1}]^{\mathrm{T}}$ that solves the system:

$$\mathbf{M}(:, 1:m-1)\boldsymbol{\beta} = \mathbf{M}(:, m). \tag{41}$$

Then, one can easily see that multiplying equation (40) by the solution vector $\boldsymbol{\beta}$ from left side provides:

$$\begin{aligned} \mathbf{c}(1:m-1)\boldsymbol{\beta} &= \mathbf{b}\mathbf{M}(:, 1:m-1)\boldsymbol{\beta} \\ &= \mathbf{b}\mathbf{M}(:, m) \\ &= c_m. \end{aligned}$$

where the forged token is expressed as a linear combination of the released tokens. Before analyzing the solvability of system (41), let us summarize the impersonation attack as follows. Consider a non-shareholder aims to impersonate the shareholder U_i by forging a valid token at point x_m. He first announces the target shareholder's identity x_m, and waits to receive the released tokens of other participating shareholders. Then, he constructs system (41) and explores for a solution. Finally, combining the gathered tokens according to the solution vector $\boldsymbol{\beta}$ produces the forged token. Note that, the forged token is obtained using public information, while there is no need to recover the coefficients of the bivariate polynomial.

In order for the impersonation attack to work, there must be a solution for system (41). The following theorem provides the solvability analysis of system (41).

Theorem 4. System $\mathbf{M}(:, 1:m-1)\boldsymbol{\beta} = \mathbf{M}(:, m)$ (41) is always solvable for every $m - 1 \geq t + 1$.

Proof: The system is solvable if $\mathbf{M}(:, m) \in C(\mathbf{M}(:, 1:m-1))$. To check this, we construct the augmented matrix which is clearly $\mathbf{M}$. From the result of theorem (1), we know that any $t+1$ column of $\mathbf{M}$ are linearly independent, thus, as long as $m - 1 \geq t + 1$, the column space is filled and:

$$rank(\mathbf{M}) = rank(\mathbf{M}(:, m-1)) = t + 1$$

Thus, satisfying $m - 1 \geq t + 1$ makes sure that the system is solvable. □

According to the solvability analysis of system (41), a non-shareholder needs to gather at least $t + 1$ valid tokens to impersonate a shareholder.

The impersonation attack is extensible. Assume that there are m_1 legitimate shareholders, $\{U_1, U_2, ..., U_{m_1}\}$, and a non-shareholder who goals to impersonate shareholders $\{U_1, U_2, ..., U_{m_2}\}$, while $t < m = m_1 + m_2 \leq t(t+1)/2$. In this case, he can construct the system:

$$\mathbf{M}(:, 1:m_1)\boldsymbol{\Gamma} = \mathbf{M}(:, m_1 + 1:m_2), \tag{42}$$

where columns of matrix $\boldsymbol{\Gamma}$ give the appropriate combination of recieved tokens to impersonate each targeted shareholder

$$\begin{aligned} \boldsymbol{\Gamma} &= \begin{bmatrix} \boldsymbol{\gamma}_1, & \cdots, & \boldsymbol{\gamma}_{m_2} \end{bmatrix} \\ &= \begin{bmatrix} \gamma_{1,1} & \cdots & \gamma_{1,m_2} \\ \gamma_{1,2} & \cdots & \gamma_{2,m_2} \\ \vdots & \vdots & \vdots \\ \gamma_{1,m_1,} & \cdots & \gamma_{m_1,m_2} \end{bmatrix}. \end{aligned} \tag{43}$$

Then, the forged tokens can be obtained as following:

$$\mathbf{c}(m_1 + 1:m_2) = \mathbf{c}(1:m_1)\boldsymbol{\Gamma}. \tag{44}$$

4.3 Discussion

The dynamic threshold protocol proposed in [9] is composed of two schemes. At the heart of the protocol, there is a pure dynamic threshold secret sharing scheme which we focused on, in this paper. At the outer layer of the protocol, there is a trivial and inefficient key establishment scheme which is suggested to be used for keeping the released tokens confidential. In more details, in this key establishment, all the m tokens should be encrypted for all the $m-1$ other participants, using $m-1$ different pairwise keys, which imposes an enormous communication and computation overheads to the participants.

We focused on the pure dynamic threshold scheme as the main contribution of [9] and challenged its claim of the unconditional security by proposing the linear subspace cryptanalysis.

Therefore, according to our results, the hybrid protocol in [9], composed of this new secret sharing scheme and this key establishment scheme is somehow equivalent the combination of the basic Shamir's scheme with this key establishment scheme. So, it is nothing more than a basic secret sharing scheme with mutually encrypted tokens, without any dynamic property.

5 CONCLUSION

The linear subspace cryptanalysis has been recently proposed for security analysis of some Shamir's secret sharing based schemes whose security relies exclusively on hiding the polynomial(s) coefficients.

The security of the dynamic threshold secret sharing scheme proposed by Harn in [9], relies on the same assumption. Benefiting from the linear subspace cryptanalysis, we challenged the unconditional security claim of this scheme by introducing methods to derive the secret and any arbitrary token in polynomial time. The proposed attacks does not relay on any special assumption except that a number of $t+1$ valid tokens should be available to the non-shareholder, which is a reasonable assumption, considering the interval given in [9] for the number of participating shareholders

as $(t, 1 + t(t + 1)/2]$. In other words, we showed that the information leaked by any $t+1$ tokens is sufficient to derive the secret, regardless of the updated value agreed for the threshold. Besides, this information is sufficient for forging any valid token.

All the above results come from this fact that the tokens vector lies in a $t + 1$ dimensional subspace regardless of how many shareholders are participating, for $m \geq t+1$. So, our claims are proved by the belongingness of the unknown values (secret and the forged tokens) to the linear subspace spanned by the known $t + 1$ valid tokens.

Finally, we emphasize that in the proposed approach there is no need to retrieve the unknown coefficients of the bivariate polynomial, whereas, the secret or the forged tokens would be constructed as a linear combination of the available valid tokens. Thus, this paper shows again that protecting the polynomial does not ensure the security of the scheme, any more.

REFERENCES

[1] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *AFIPS*, 1979, pp. 313-317.

[3] K. Martin, J. Pieprzyk, R. Safavi-Naini, and H. Wang, "Changing thresholds in the absence of secure channels," in *Proc. ACISP* (Lecture Notes in Computer Science), Springer Nature, vol. 1587, 1999, pp. 177-191.

[4] Z. Zhang, Y. M. Chee, S. Ling, M. Liu, and H. Wang, "Threshold changeable secret sharing schemes revisited," *Theoretical Computer Science*, vol. 418, pp. 106-115, 2012.

[5] H. Wang and D. S. Wong. "On secret reconstruction in secret sharing schemes," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 473-480, Jan. 2008.

[6] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7195-7206, Dec. 2016.

[7] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," *Proc. 28th Annu. Symp. Found. Comput. Sci.*, pp. 427-438, 1987.

[8] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Proc. Annu. Int. Cryptol. Conf. (CRYPTO)*, pp. 129-140, 1992.

[9] L. Harn and C. F. Hsu, "Dynamic threshold secret reconstruction and its application to the threshold cryptography," *Inf. Process. Lett.*, vol. 115, no. 11, pp. 851-857, 2015.

[10] L. Harn, "Group authentication," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1893-1898, Sep. 2013.

[11] L. Harn, "Secure secret reconstruction and multi-secret sharing schemes with unconditional security," *Secur. Commun. Netw.*, vol. 7, no. 3, pp. 567-573, 2014.

[12] J. Li, M. Wen, and T. Zhang, "Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A Networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 408-417, Jun. 2016.

[13] Z. Ahmadian and S. Jamshidpour, "Linear subspace cryptanalysis of a secret sharing-based group authentication scheme," *IEEE Trans. Inf. Forens. Security*, in press.

[14] C. G. Khatri and C. R. Rao, "Solutions to some functional equations and their applications to characterization of probability distributions," *The Indian J. Stat. Series A*, vol. 30, no. 2, pp. 167-180, 1968.

[15] S. Banerjee and A. Roy, *Linear Algebra and Matrix Analysis for Statistics, Texts in Statistical Science*. CRC Press, 2014.