IJICTR International Journal of Information & Communication Technology Research

Volume 7- Number 4- Autumn 2015 (35-42)

Efficient Verifiable Dynamic Threshold Secret Sharing Scheme Based on Elliptic Curves

Amir Alaei
Engineering Department of ICT
Imam Hussein Comprehensive University
Tehran, Iran
alaei@saba.org.ir

Mohammad H. Tadayon
Iran Telecommunication Research Center (ITRC)
Tehran, Iran
tadayon@itrc.ac.ir

Received: May 17, 2015- Accepted: August 5, 2015

***Abstract*—A dynamic threshold secret sharing (DTSS) scheme allows the secret to be updated without changing the shares. The first DTSS scheme was proposed by Laih *et al.* in 1991. Several other schemes based on different methods have been proposed since then. In 2007, Chen *et al.* proposed a verifiable DTSS scheme based on elliptic curves and bilinear maps, which is almost efficient. In this paper, we propose an alternative verifiable DTSS scheme using elliptic curves and bilinear maps. The proposed scheme is computationally secure, and the secret and/or threshold parameter can change to any arbitrary values multiple times. Furthermore, in our scheme, there is no secure channel and participants do not need to save any information or extra shares ahead of time. Since the running time is an important factor for practical applications, we provide a complexity comparison of our approach with respect to Chen *et al.*'s scheme. The comparison between the proposed scheme and that of Chen *et al.* indicates that the new scheme is more efficient, that it means, it has much lower computational complexity, as well as smaller storage requirements.**

Keywords- Dynamic threshold secret sharing; Elliptic curve; Bilinear pairing; Verifiable; Computational security

I. INTRODUCTION

One way to provide both secrecy and availability for a given secret (highly sensitive information) is to employ secret sharing schemes. A secret sharing scheme is a method of distributing a secret among a set of participants (shareholders) by giving each participant a share (shadow) in such a way that only authorized subsets of participants (defined by the access structure Γ) can reconstruct the secret from pooling their shares, but any unauthorized subset of them cannot. Specifically, in a (t, n)-threshold secret sharing (TSS) scheme, a secret s is distributed as shares among n participants in such a way that any group of at least t participants can recover the secret s, while no groups having at most $t - 1$ participants can uniquely determine the secret s.

In 1979, Shamir [23] and Blakley [2] independently found practical solutions to (t, n)-threshold secret sharing schemes, so as to facilitate the distributed storage of secret information in an unsafe environment. Shamir's threshold scheme is based on polynomial interpolation over a finite field. Despite introducing other secret sharing schemes, for instance [1] and [10], Shamir's scheme has received more attention than the others, owing to its effective applicability.

Secret sharing schemes are highly versatile cryptographic primitives and have been employed in various applications, such as protection of cryptographic keys, access control, key recovery mechanisms, e-voting, ad hoc networks, secure multiparty computation, to mention but a few.

The security of a threshold scheme is categorized nto two levels: information theoretical (perfect) security and computational security. A (t, n)-threshold secret sharing scheme is called perfect if any subset of less than t participants neither can reconstruct the secret, nor obtain any information on it. It has been shown that in perfect secret sharing schemes, the size of each share must be at least the same as the secret's [25]; in the case of equality, the scheme is called ideal. A (t, n)-threshold secret sharing scheme is called computationally secure if for any subset of less than t participants, it is computationally infeasible to reconstruct the secret s in polynomial time [14].

Now, suppose that the secret is kept unchanged in the scheme for a long period of time and the adversary's capabilities increase over time, for example by compromising more participants, or the number of colluding participants increases over time. Therefore, the adversary or the colluding participants may finally obtain the secret. One approach to address the issue can be by increasing the threshold value. Other solutions to tackle this problem have been proposed in the literature, but they either have large storage requirements, or they are limited to a predefined threshold modification or they require a secure channel between the dealer and the participants or between each pair of participants [20].

In classic threshold schemes, when the secret or threshold is changed, the corresponding shares must be regenerated and then secretly distributed to participants again. This is inefficient due to the overhead in the generation and distribution of shares, especially when the number of the shares is large [27].

In 1991, Laih *et al.* [15] introduced the concept of dynamic threshold secret sharing (DTSS) scheme in order to resolve the above mentioned issue. A DTSS scheme allows the secret to be renewed and/or the threshold parameter to be changed, while the originally distributed shares remain unchanging. DTSS schemes usually require a number of public values. The participant who wishes to participate in the secret reconstruction process, derives the corresponding pseudo-share from his/her master-share and these public values.

In this paper, we propose a verifiable DTSS scheme with some desirable features as follows:

- Each participant holds only one permanent, private share, which is chosen by himself/herself. Moreover, the proposed scheme requires no secure channels and consequently the cost of the scheme can be reduced.
- It has the minimum storage cost, because participants do not need to store any information or extra shares ahead in order to change the secret or threshold later.
- It is flexible since the threshold can be changed to any arbitrary values multiple times.
- The combiner can detect and identify dishonest participants just before secret reconstruction process. This feature does not allow the cheaters to participate in the reconstruction process; So, the cheaters can not prevent the correct secret reconstruction.
- In our scheme, the public values are independent of the number of changes in the secret and/or threshold value.

The proposed scheme is computationally secure. More precisely, the security of the proposed scheme as [5] relies on the intractability of the elliptic curve discrete logarithm problem (ECDLP). Regarding the security model, computational security is theoretically weaker than information-theoretical (perfect) security [4] but computational security is not a practical limitation at all. In fact, most implementations of perfect secret sharing schemes result in actual computational security [14].

We also note that various efficiency measurements of techniques for access structure change have been proposed, which tend to be measures of either [16]:

1) The amount of secret information that participants need to store.
2) The amount of secret information that participants need to communicate as a part of the structure change.
3) The amount of public information needed to be broadcast to facilitate a structure change.

The computational complexity and the number of public values are two important factors for evaluating the efficiency of DTSS schemes. Some publications such as [5], [26] appeared to reduce the value of these parameters.

The authors compare the new verifiable DTSS scheme with that of Chen *et al.* [5]. This comparison shows that the new scheme reduces the computational complexity and the size of public values, while security features remain the same.

The remainder of this paper is organized as follows. In the next section, we recall the polynomial interpolation problem and the concepts of elliptic curves as well as bilinear maps, since they have a major role in our construction. In Section III, we briefly review Chen *et al.*'s scheme. In Section IV, we describe our verifiable DTSS scheme. A thorough analysis of the proposed scheme together with a comparison between the proposed scheme and the constructions from [5] is made in Section V. Finally, Section VI concludes the paper.

II. PRELIMINARIES

In this section, we recall three problems which have major roles in proving the correctness and efficiency of the scheme described in Section IV.

A. Points Interpolation and Polynomial Evaluation

Suppose that we are given $n + 1$ points (x_0, y_0), ..., (x_n, y_n) such that the x_i's are distinct in a field K. The Lagrange interpolating polynomial $f(x)$ is the only polynomial of degree at most n passing through the above $n + 1$ points. Algorithm 4.6.1 from [8] computes the $n + 1$ coefficients of $f(x)$ using $3n(n + 1)/2$ field additions, $n(n + 1)$ field multiplications, $n(n + 1)/2$

field inversions in K.

Now, let $f(x)$ be a polynomial of degree n over K. Using Horner's method, one can efficiently evaluate a point of $f(x)$ by n field multiplications and n field additions.

B. Elliptic Curves

Let F_q be a finite field of $q = p^m$ elements, where p is the characteristic of F_q. We consider separately the cases where the underlying field F_q has characteristic different from 2 and 3, or has characteristic equal to 2 or 3 [12], [9].

1. If F_q is a field of characteristic not equal to 2 and 3, i.e., $p > 3$, then an elliptic curve E over F_q is the set of all points (x, y) with $x, y \in F_q$ which satisfy the equation

$$E : y^2 = x^3 + ax + b$$

together with an extra point O, called the point at infinity, where the constants $a, b \in F_q$ and the condition $\Delta = 4a^3 + 27b^2 \neq 0$. The condition Δ is called the discriminant of E.

2. If F_q is a field of characteristic 2, then an elliptic curve E over F_q is the set of all points (together with a point at infinity O) which satisfy an equation of the two forms either

$$y^2 + xy = x^3 + ax^2 + b, \quad (1)$$

or

$$y^2 + cy = x^3 + ax + b. \quad (2)$$

An elliptic curve defined by (1) is said to be non-supersingular and has discriminant $\Delta = b \neq 0$, while one which is defined by (2) is said to be supersingular and has discriminant $\Delta = c^4 \neq 0$.

3. If F_q is a field of characteristic 3, then an elliptic curve E over F_q is the set of all points (together with a point at infinity O) which satisfy an equation of type either

$$y^2 = x^3 + ax^2 + b, \quad (3)$$

or

$$y^2 = x^3 + ax + b. \quad (4)$$

An elliptic curve defined by (3) is said to be non-supersingular and has discriminant $\Delta = -a^3b \neq 0$, while one which is defined by (4) is said to be supersingular and has discriminant $\Delta = -a^3 \neq 0$.

The condition $\Delta \neq 0$ ensures that the elliptic curve is nonsingular (or smooth), that is, there are no points at which the curve has two or more distinct tangent lines.

Elliptic curve point addition is defined according to the "chord-tangent process," and involves a few arithmetic operations in F_q. Under this addition, the points of $E(F_q)$ form an abelian group, with the point O serving as its identity element. By Hasse's theorem, the order of the group is $q + 1 - t$, where $|t| \leq 2\sqrt{q}$. The type of the group is (n_1, n_2), i.e., $E(F_q) \simeq Z_{n_1} \oplus Z_{n_2}$, where $n_2 | n_1$, and furthermore $n_2 | q - 1$ [17]. For all elliptic curves over finite fields, the group is always finite and it is also highly likely to be cyclic (or almost cyclic) [24].

In 1985, Miller [18] and Koblitz [11] independently proposed the idea of using elliptic curves in public-key cryptography. Elliptic curve cryptosystem (ECC) provides the same level of security as RSA or discrete logarithm (DL) cryptosystems with substantially shorter operands (approximately 160-256 bits vs. 1024-3072 bits). In many cases, ECC has performance advantages (fewer computations) and bandwidth advantages (shorter keys and signatures) over RSA and discrete logarithm schemes.

It should be stressed that this security is only achieved if cryptographically strong elliptic curves are used. There are several families of curves that possess cryptographic weaknesses, e.g., supersingular curves. To avoid the reduction algorithms from [17], [7], the curve should be non-supersingular. Hence, if a supersingular elliptic curve is desired in practice, then it should be carefully chosen.

Let E be an elliptic curve over the finite field F_q, and suppose P be a point with order m on the elliptic curve E where m is large (for example, $m > 2^{160}$), and Q is some other point on the same curve. The elliptic curve discrete logarithm problem (ECDLP) is the problem of finding the integer $k \in Z_m$ such that $Q = kP$, provided that such an integer exists. We call k the elliptic discrete logarithm of Q with respect to P. There is no probabilistic polynomial time algorithm (in $\log_2 q$) for solving ECDLP [22]. Of course, this statement assumes a well-chosen elliptic curve [24].

The Pohlig-Hellman algorithm reduces the determination of k to the determination of k modulo each of the prime factors of m. Hence, in order to achieve the maximum possible security level, m should be prime [13]. The best algorithm known until now to solve an ECDLP is the Pollard's rho method, which compute an elliptic curve discrete logarithm with an average of $O(\sqrt{n})$ steps, where a step here is an elliptic curve addition. Therefore, this is a completely exponential algorithm. Since determining elliptic curve discrete logarithms is harder than in the case of multiplicative groups of finite fields, one can use smaller elliptic curve groups while maintaining the same level of security [24].

The elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken [22].

C. Bilinear Maps

Let $G = \langle P \rangle$ be a cyclic additive group of an elliptic curve E generated by P whose order is a prime number q. We define the following problems for all $a,b,c \in Z_q^*$:

Definition 1. *The elliptic curve Diffie-Hellman problem (ECDHP) is the problem of computing the value of abP from the known values of P, aP and bP [9]. Clearly, the ECDHP reduces to the ECDLP in polynomial-time.*

Definition 2. *The elliptic curve decision Diffie-Hellman problem (ECDDHP) is the problem of determining whether cP = abP or not [9].*

Now, we consider a cyclic additive group $G_1 = \langle P \rangle$ and a cyclic multiplicative group G_2. These two groups are assumed to have the same large prime order q. We also assume that the ECDDHP in G_1 is easy, while the DDHP in G_2 is hard, and both the ECDHP in G_1 and the discrete logarithm problem (DLP) in G_2 are hard. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties [19], [3]:

1. The map e is bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and any $a, b \in Z_q$.
2. $e(.,.)$ is not degenerate: $e(P, P) = 1_{G_2}$, where 1_{G_2} is the identity element of G_2.
3. There exists a computationally efficient algorithm to compute $e(P, Q) \in G_2$ for all $P, Q \in G_1$.

III. Chen Et Al.'s DTSS Scheme

In this section, we briefly explain the DTSS scheme proposed by Chen *et al.* [5]. We can divide Chen *et al.*'s scheme into four phases: system setup, secret distribution, secret recovery and secret redistribution. Each participant U_i ($i = 0, 1, ..., n-1$) selects his/her own private share by himself/herself, which can be used repetitively in various secret sharing schemes. In this scheme, the dealer publishes related public information and only the dealer can modify the published information, whereas the others can only read or download it.

A. System Setup

Let G_1 be a cyclic additive subgroup of order q (that q is a large prime number) and G_2 be a multiplicative group of non-zero elements of order q. Suppose that $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map. The dealer chooses a generator P of G_1, and a cryptographic hash function $h : G_1 \rightarrow Z_q^*$, then publishes q, G_1, G_2, e, P, h on a public bulletin. Each participant U_i ($i = 0, 1, ..., n-1$) randomly selects a private share $s_i \in Z_q^*$, computes the public share $P_i = s_iP$, and then submits P_i to the dealer. The dealer verifies whether $P \neq P_i \neq P_j$ ($i \neq j$) in order to keep different participants from using the same private share, and then publishes P_i's ($i = 0, 1, ..., n-1$) on the public bulletin.

B. Secret Distribution

In this stage, the dealer chooses the secret s, computes and publishes some public values. Then, the dealer does the following steps.

Randomly pick an $r \in Z_q^*$, compute the secret $s = h(rP) \in Z_q^*$, check whether $sP \neq P_i$ ($i = 0, 1, ..., n-1$) and then publish the value of sP.

Choose the threshold value t, randomly pick a generator g of Z_q^*, and form an $(n + 1 - t) \times (n + 1)$ matrix M, where $n < q - 1$,

$$M = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & g & \dots & g^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & g^{n-t} & \dots & g^{n(n-t)} \end{pmatrix}.$$

Compute sP_i ($i = 0, 1, ..., n-1$), form an $(n + 1)$ column vector $A = (rP, sP_0, ..., sP_{n-1})^T$, where T represents the transpose of the vector A, and compute the $(n + 1 - t)$ column vector V,

$$V = MA = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & g & \dots & g^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & g^{n-t} & \dots & g^{n(n-t)} \end{pmatrix} \begin{pmatrix} rP \\ sP_0 \\ \vdots \\ sP_{n-1} \end{pmatrix} = \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_{n-t} \end{pmatrix}. \quad (5)$$

Finally, publish g and C_i ($i = 0, 1, ..., n-t$).

C. Secret Recovery

The equation (5) is the system of $n + 1 - t$ linear equations in $n + 1$ unknown elements of G_1. Clearly, if t participants submit their shares as s_isP, then the combiner can obtain $n + 1 - t$ linear equations in $n + 1 - t$ unknowns. Therefore, other $n + 1 - t$ unknowns will be revealed, including rP. Consequently, the secret s can be recovered as $s = h(rP)$. Note that any $(n + 1 - t) \times (n + 1 - t)$ sub-matrices of M is full-rank, thus (5) has a unique solution over the group G_1 .

D. Secret Redistribution

The dealer chooses a new threshold value t', a new secret s', and an $r' \in Z_q^*$. The dealer then proceeds as above secret distribution phase. Finally, he/she computes new public information from participants' public shares, and publishes the new public information.

IV. The Proposed Scheme

Here, we propose a verifiable dynamic threshold secret sharing scheme using elliptic curves and bilinear maps. The proposed scheme provides resistance against cheating by malicious participants and reduces the size of the public values as well as the computational complexity with respect to [5].

The proposed scheme consists of four phases: (1) initialization, (2) secret distribution, (3) share verification and secret reconstruction, and (4) secret redistribution. Throughout this section, we denote the n participants by $U_1, U_2, ..., U_n$ and the honest dealer by D who is available during the initialization and running phases, but only has access to an authenticated public broadcast channel, on which information is transmitted instantly and accurately to all participants.

Let q be a sufficiently large prime number (for example, q should be at least 160 bits long) and $G = \langle P \rangle$ be a cyclic additive group of order q, that P is a generator of G. In addition, suppose Z_q is the finite field of integers modulo q, and distinct nonzero values $x_1, x_2, \ldots, x_n \in Z_q$ are the participants' identifiers, as well as $h : \{0,1\}^* \rightarrow Z_q$ is a hash function mapping a binary string of arbitrary length to an element of Z_q.

A. Initialization

First of all, the dealer selects a generator $P \in G$, and publishes the value of P. Then, each participant U_i, $1 \le i \le n$, selects a random integer $s_i \in Z_q^*$ as his/her private share, computes $P_i = s_iP$ as his/her corresponding public share and then sends it to the dealer through a public channel broadcast message.

Note that $P_i(= s_iP)$ is an element of the group G (and correspondingly is a point on an elliptic curve *E)* and it is computed by adding P to itself s_i times.

On receiving all public shares P_i's of n participants, the dealer should ensure $P_i \ne P_j \ne P$ for every distinct i and j, $1 \le i \ne j \le n$. Once $P_i = P_j$ for some distinct i and j, those participants should be demanded to choose different private shares until all P_i's are distinct for $i = 1, 2, ..., n$. Finally, the dealer publishes all P_i's ($i = 1, 2, ..., n$).

B. Secret Distribution

Here, the dealer performs the following steps.

1. The dealer secretly chooses a random integer $r \in Z_q^*$ and publishes the public value of $Q = rP$.
2. The dealer computes pseudo-shares rP_i as well as $h(rP_i)$ for $i = 1, 2, ..., n$. Having $n + 1$ points $(0, s), (x_{i_1}, h(rP_{i_1})), \ldots, (x_{i_n}, h(rP_{i_n}))$ using Lagrange interpolation formula, the dealer forms a random polynomial $f(x) \in Z_q[x]$ of degree at most n.

$$f(x) = s + a_1x + a_2x^2 + \ldots + a_nx^n \pmod q.$$

3. Finally, he/she chooses the $n - t + 1$ smallest integers $d_1, d_2, ..., d_{n-t+1} \in Z_q^* \setminus \{x_i \mid i = 1, 2, ..., n\}$, computes and publishes $f(d_1), f(d_2),..., f(d_{n-t+1})$.

C. Share Verification and Secret Reconstruction

One of the significant advantages of our scheme is that the combiner is able to verify the validity of the pseudo-shares by using bilinear maps. Suppose at least t participants $U_{i_1}, U_{i_2}, \ldots, U_{i_t}$ submit their pseudo-shares $s_{i_1}Q, s_{i_2}Q, \ldots, s_{i_t}Q$ to the combiner. On receiving $s_{i_k}Q$ ($k = 1, 2, ..., t$), the combiner (who may be one of the participants) first verifies the validity of the submitted pseudoshares by checking whether $e(s_iQ, P) = e(Q, P_j)$ for each of the participants who participate in the secret reconstruction process.

Next, the combiner computes $h(s_{i_k}Q)$ for $k = 1, 2, ..., t$. Having the t values $h(s_{i_1}Q), \ldots, h(s_{i_t}Q)$ as well as the $n - t + 1$ public values $f(d_1), ..., f(d_{n-t+1})$ and using Lagrange interpolation formula, the combiner is able to recover the secret s.

We also remark that each participant U_i can compute the pseudo-share rP_i from the public value Q and his/her private share s_i, since:

$$s_iQ = s_irP = rs_iP = rP_i .$$

D. Secret Redistribution

The dealer chooses a new secret s' and/or a new threshold t', as well as a new random value $r'(\ne r)$. The dealer then proceeds as secret distribution phase and finally publishes the new public values $Q'(= r'P)$, $f'(d_1),..., f'(d_{n-t'+1})$.

A comprehensive analysis of the proposed verifiable DTSS scheme and a comparison with the construction from [5] is presented in the next section.

V. Investigation of the Proposed Scheme

In this section, we discuss the security and performance of the proposed scheme in two parts. In the first part, it is shown that the scheme provides computational security. In the second part, efficiency of the scheme is investigated and a comparison with [5] is made. The reason behind this choice of the reference scheme is due to the simplicity of its structure. The comparison results show that the proposed scheme is more efficient, i.e., it has lower computational complexity and reduces the size of the public values, while preserving the same security features.

A. Security Analysis

So as to demonstrate that the proposed scheme provides computational security, we state the three following theorems.

Theorem 1. *In the proposed scheme, any subset of participants whose number is less than the corresponding threshold value t, obtain no information (from a computational security point of view) about the related secret s.*

Proof. To prove this assertion, suppose there exist at most $t - 1$ participants who conspire to determine the secret s. To achieve this goal, the colluders have to obtain $n + 1$ points of $f(x)$ (as defined in 4.3). However, they have at most n points of it, that is, $t - 1$ points $(x_{i_1}, h(s_{i_1}Q)), \ldots, (x_{i_{t-1}}, h(s_{i_{t-1}}Q))$ and the $n - t + 1$ public points $(d_1, f(d_1)), ..., (d_{n-t+1}, f(d_{n-t+1}))$. The secret polynomial coefficients tend to be randomly and uniformly distributed modulo q (This is not a theorem, but it is an experimentally observed fact.), since $f(x)$ was first constructed by using $n + 1$ points which were chosen at random by the dealer and each of the n participants. Hence, the secret s takes all the values in Z_q with the equal probability when the unknown point of $f(x)$ varies over Z_q and, as a consequence, the colluders obtain no information about the secret. On the one hand, it is computationally infeasible to compute the value of the private share s_i from the two known public values $P_i(= s_iP)$ and P, due to the difficulty of the ECDLP in G_1. Moreover, it is computationally infeasible to compute the value of s_i by reducing the ECDLP in G_1 to an instance of the DLP in G_2 by using a bilinear map $e(P_i, P) = e(P, P)^{s_i} \in G_2$, due to the difficulty of the DLP in G_2. On the other hand, it is computationally infeasible to compute the value of $rP_i(= s_iQ)$ from the known public values P, $P_i(= s_iP)$ and $Q(= rP)$, due to the difficulty of the ECDHP in G_1. Therefore, the public values leak no information in polynomial-time about the private shares or pseudo-shares of the non-colluding participants.

□

Theorem 2. *In the proposed scheme, after changing r to r', the threshold value from t to t'(> t) (the secret is not changed), and updating the old public values Q(= rP), f(d1), .. ., f(dn–t+1) to the new ones Q'(= r'P), f'(d1), .. ., f'(dn–t'+1), there is no information leakage from the old public values to the secret.*

Proof. Because all coefficients of the new secret polynomial $f'(x)$ of degree n (including the secret) tend to be randomly and uniformly distributed over Z_q, the new public values $f'(d_1),..., f'(d_{n-t'+1})$ generated by it are independent of those generated by the old secret polynomial $f(x)$ of degree n. Thus, there is no information leakage from the old public values to the secret.

Theorem 3. *The shares provided by the participants during the secret reconstruction phase can be verified so that cheaters are identified.*

Proof. Suppose that the participant U_i submits his/her pseudo-share s_iQ to the combiner. As mentioned earlier, it is computationally infeasible for an adversary to compute $s_iQ(= rP_i)$ from the known public values P, $P_i(= s_iP)$ and $Q(= rP)$. Hence, only the dealer and the participant U_i are able to compute the value of $s_iQ(= rP_i)$. On receiving s_iQ, the combiner employs a bilinear pairing and verifies whether $e(s_iQ, P) = e(Q, P_i)$ or not. Now, suppose that a participant U_j cheats during the secret reconstruction process, thus he/she must submit an invalid value s_j^*Q to the combiner (since P, Q and P_j are public and known, therefore the only way for cheating is to change the value s_jQ to a different value s_j^*Q), and the combiner will run the verification algorithm and verify whether $e(s_j^*Q, P) = e(Q, P_j)$ or not; But $e(s_j^*Q, P) \neq e(Q, P_j)$, since:

$$e(s_j^*Q, P) = e(Q, P)^{sj*} = e(Q, s_j^*P) \neq e(Q, P_j)$$

hence, the cheater U_j will be easily identified.

The three above theorems ensure that the proposed scheme provides the desired level of security.

B. *Efficiency Comparison*

As mentioned in Section I, the computational and storage costs represent crucial factors taken into account when implementing a protocol as a part of a commercial application. Here, we study the cost of our construction and compare the proposed scheme with the scheme of [5] from the following points of view: the size of public values' storage and the computational complexity of the schemes, as well as the security features. We assume that picking random elements from the sets F_q and G_1 has a negligible computational cost.

Table I defines the notations used in this subsection. Using the approach of [6], computation of the inverse of an $n \times n$ Vandermonde matrix requires $5n(n-1)/2$ field multiplications, n^2 field divisions (i.e., n^2 field inversions and n^2 field multiplications), and $5n(n-1)/2$ field additions. The time complexity of various operations in terms of time complexity of a field multiplication is illustrated in Table II which is extracted from [13], [6], [21]. The values in both columns of Table II belong to F_q with $q \approx 2^{160}$. We assume that picking random elements from the sets F_q, G and G_1 has a negligible computational cost.

TABLE I. DEFINITION OF GIVEN NOTATIONS

Notations	Definitions
T_m	Time complexity for computing a field multiplication
T_{add}	Time complexity for computing a field addition
T_{inv}	Time complexity for computing a field inversion
$T_{ec\text{-}add}$	Time complexity for computing an elliptic curve addition or doubling
$T_{ec\text{-}mul}$	Time complexity for computing kP
T_{int}	Time complexity for interpolating $n + 1$ points
$T_{inv\text{-}Van}$	Time complexity for computing the inverse of an $n \times n$ Vandermonde matrix
T_h	Time complexity for executing a hash function

The comparison results between the proposed scheme and [5] are illustrated in Table III. From the results, it is easy to infer that the size of public values in our scheme is smaller than [5]. The required computational cost for both schemes has been estimated by accumulating execution times of all the required operations in terms of T_m.

Let $size(x)$ denote the number of bits used to represent the natural integer x. We have $size(x) = \lfloor \log_2 x \rfloor + 1$. We also remark that a point P on an elliptic curve $E(F_q)$ in affine coordinates is represented as (x_P, y_P), so the size of P is equal to $2(\lfloor \log_2 q \rfloor + 1)$. As a consequence, the size of our public elements represents a total of $(3n - t + 6)(\lfloor \log_2 q \rfloor + 1)$ bits, while the size of public elements in Chen *et al.*'s scheme is $(4n - 2t + 8)(\lfloor \log_2 q \rfloor + 1)$ bits. Hence, a priori, our technique provides significant size benefit.

The dealer in the proposed scheme utilizes Lagrange interpolation formula for constructing the secret polynomial $f(x)$ and then evaluates $n - t + 1$ points of it together with only a very few computations over an elliptic curve group for generating the public values except the public shares. However in the Chen *et al.*'s scheme, the dealer employs multiplication of two matrices together with almost all computations over an elliptic curve group for generating the public values except the public shares. On the other hand, the combiner in our scheme has to use Lagrange interpolation formula (over a finite field) for recovering the secret s, while in Chen *et al.*'s scheme, the combiner has to solve a system of $n - t + 1$ linear equations in $n - t + 1$ unknowns (over an elliptic curve group) in order to reconstruct the secret s. Clearly, Lagrange interpolating is much simpler than simultaneously solving linear equations [28]. The security of the proposed scheme is the same as [5], that is, both of them are based on the difficulty of the ECDLP. As a final point, we remark a drawback of Chen *et al.*'s scheme, that is to say, the dealer in Chen *et al.*'s scheme is not able to change the threshold value t to $t'(> t)$ without changing the secret, while the dealer in our scheme is able to do. Therefore, the proposed scheme is more efficient and can be widely used in practice.

TABLE II. UNIT CONVERSION OF VARIOUS OPERATIONS IN TERMS OF T_m

Time complexity of operation units	Time complexity in terms of a field multiplication
T_{add}	≈ 0
T_{inv}	$3T_m$
$T_{ec\text{-}add}$	$5T_m$
$T_{ec\text{-}mul}$	$1200T_m$
T_{int}	$(5n(n+1)/2)T_m$
$T_{inv\text{-}Van}$	$(5n(n-1)/2+4n^2)T_m$
T_h	T_m

TABLE III. COMPUTATIONAL COMPLEXITY OF THE TWO VERIFIABLE DTSS SCHEMES

	Proposed Scheme	Chen et al.'s Scheme [5]
Size of each private share (in bits)	$\lfloor \log_2 q \rfloor + 1$	$\lfloor \log_2 q \rfloor + 1$
Public values' size (in bits)	$(3n - t + 6)(\lfloor \log_2 q \rfloor + 1)$	$(4n - 2t + 8)(\lfloor \log_2 q \rfloor + 1)$
Size of renewed public values (in bits)	$(n - t + 3)(\lfloor \log_2 q \rfloor + 1)$	$(2n - 2t + 4)(\lfloor \log_2 q \rfloor + 1)$
Computational complexity at the dealer	$n(3.5n - t + 1204.5)T_m + 1200T_m$	$1206n(n - t)T_m + 1205nT_m + 2401T_m$
Computational complexity at the combiner	$2.5n(n + 1)T_m + tT_m$	$1205(t + 1)(n - t)T_m + 4(n - t + 1)^2T_m + 2.5(n - t + 1)(n - t)T_m + n(n - t)T_m + 5tT_m + 1201T_m$

VI. CONCLUSION

In this paper, we proposed a verifiable dynamic threshold secret sharing scheme allowing the secret and/or the threshold parameter to be changed over an insecure network without any changes in the private shares. Like [5], the security of our scheme is based on the ECDLP, due to this, our scheme requires no secure channels, and due to the employment of a bilinear pairing, the combiner is able to verify the validity of the pseudo-shares in the secret reconstruction process. Furthermore, the storage requirements of our public values are much smaller than [5]. To the best of our knowledge and compared to existing methods in the literature, our scheme is more efficient, that is, it requires smaller public values and has lower computational complexity, while preserving the desired security features.

VII. ACKNOWLEGEMENT

This work was partially supported by Iran Telecommunication Research Center (ITRC) under the grant No. T/500/8999.

REFERENCES

[1] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 208–210, Mar. 1983.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. AFIPS 1979 National Computer Conf.*, vol. 48, pp. 313–317, 1979.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *21st Annual International Cryptology Conference*. pp. 213-229, Springer-Verlag, Aug. 2001.

[4] C. Cachin, "On-line secret sharing," in *Proc. 5th IMA Conf. Cryptography and Coding (Lecture Notes in Computer Science)*, vol. 1025, Springer-Verlag, pp. 190–198, 1995.

[5] W. Chen, X. Long, Y. B. Bai, and X. P. Gao, "A new dynamic threshold secret sharing scheme from bilinear maps," *Int. Conf. Parallel Processing Workshops (ICPPW'07)*, pp. 19–22, Sept. 2007.

[6] M. Dejnakarintra and D. Banjerdpongchai, "An algirithm for computing the analytical inverse of the vandermonde matrix," in *Proc. 3rd Asian Control Conf.* Shanghai, China, pp. 2051–2054, July 2000.

[7] G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Mathematics of Computation*, vol. 62, pp. 865–874, 1994.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations (Third Edition)*. The Johns Hopkins University Press, 1996.

[9] D. R. Hankerson, A. J. Menezes, and S. A. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2003.

[10] E. D. Karnin, J. W. Greene, and M. E. Hellman, "On secret sharing systems," *IEEE Trans. Inf. Theory*, vol. 29, no. 1, pp. 35–41, Jan. 1983.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, 1987.

[12] N. Koblitz, *A Course in Number Theory and Cryptography*. Springer- Verlag, 1994.

[13] N. Koblitz, A. J. Menezes, and S. A. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, pp. 173–193, Springer, 2000.

[14] H. Krawczyk, "Secret sharing made short," in *Proc. 13th annual int. cryptology conf. on Advances in cryptology, CRYPTO'93, LNCS 773*, pp. 136–146, 1994.

[15] C. S. Laih, L. Harn, J. Y. Lee, and T. Hwang, "Dynamic threshold scheme based on the definition of cross-product in an n-dimensional linear space," *J. Inf. Science and Engineering*, vol. 7, pp. 13–23, 1991.

[16] K. M. Martin, "Dynamic access policies for unconditionally secure secret sharing schemes," in *Proc. IEEE Inf. Theory Workshop*, pp. 61–66, Oct. 2005.

[17] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Trans. Inf. Theory*, vol. 39, pp. 1639–1646, Sept. 1993.

[18] V. Miller, "Uses of elliptic curves in cryptography," Advances in Cryptology Crypto '85 (Lecture Notes in Computer Science), vol. 218, pp. 417–426, Springer-Verlag, 1986.

[19] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Topics in Cryptology CT-RSA, Lecture Notes in Computer Science*, vol. 3376, pp. 275–292, Springer-Verlag, Feb. 2005.

[20] M. Nojoumian and D. R. Stinson, "Dealer-free threshold changeability in secret sharing schemes," *Cryptology ePrint Archive, Report 2009/268*, 2009. [Online]. Available: http://eprint.iacr.org/

[21] C. J. Mitchell, F. Piper, and P. Wild, "Digital signature," in *Contemporary Cryptology: The Science of Inf. Integrity, IEEE Press*, pp. 325–378, 1992.

[22] C. Popescu, "An identification scheme based on the elliptic curve discrete logarithm problem," in *Proc. 4th Int. Conf./Exhibition on High Performance Computing in the Asia-Pacific Region*, vol. 2, pp. 624–625, 2000.

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

[24] N. Smart, *Cryptography, An introduction*. McGraw-Hill, 2002.

[25] D. R. Stinson, "An explication of secret sharing schemes," *Designs, Codes and Cryptography*, vol. 2, no. 4, pp. 357–390, 1992.

[26] H. M. Sun and S. P. Shieh, "Construction of dynamic threshold schemes," *Electron. Lett.*, vol. 30, pp. 2023–2025, Nov. 1994.

[27] H. M. Sun and S. P. Shieh, "On dynamic threshold schemes," *Inf. Process. Lett.*, vol. 52, pp. 201–206, 1994.

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

Amir Alaei received the B.Sc. degree in Electronics Engineering with the honor degree from Islamic Azad University, Karaj branch, Alborz, Iran in 2007 and the his M.Sc. degree in Telecommunication in the field of Cryptography from IHU, Tehran, Iran in 2011. Currently, he is a security expert at Smart Grid Department of Iran Energy Efficiency Organization (IEEO), Tehran, Iran. His research interest includes: Cryptography, Information Systems Security, Network Security and Smart Grid Security.

Mohammad Hesam Tadayon received the B.Sc. degree in mathematics from the University of Mazandaran ,Babolsar, Iran, in 1995, the M.Sc. degree in mathematics from the University of Tarbiat Modarres, Tehran, Iran, in 1997, and the Ph.D. degree in applied mathematics (coding and cryptography) from the University of Tarbiat Moallem of Tehran (Kharazmi), Tehran, Iran, in 2008. He is now an Assistant Professor at the Iran Telecommunications Research Center. His research interests include error-control coding, information theory and data security.