*I.J. Information Technology and Computer Science,* **2014, 12, 18-23**
Published Online November 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2014.12.03

# Polynomial Differential-Based Information-Theoretically Secure Verifiable Secret Sharing

**Qassim Al Mahmoud**
University of Bucharest / Faculty of Mathematics and Computer Science, Bucharest, Romania
Email: qassimalmahmoud@gmail.com

***Abstract***— In Pedersen's VSS scheme the secret is embedded in commitments. And the polynomial used is of degree at most (t-1). In strong – (t, n) VSS which based on Pedersen's scheme that polynomial in verification purpose is public polynomial. The public polynomial in their scheme which acts in verification purpose is not secure. And the secret is secure if the dealer cannot solve the discrete logarithm problem. In our propose scheme we will satisfy the security requirements in strong t-consistency and consider the security on verification polynomial used. We will show in shares verification algorithm the participants can verify that their shares are consistent and the dealer is honest (i.e. the dealer cannot success in distributing incorrect shares even the dealer can solve the discrete logarithm problem.) before start secret reconstruction algorithm. The security strength of the proposed scheme lies in the fact that the shares and all the broadcasted information convey no information about the secret.

***Index Terms***— Secret Sharing, T-Consistency, Strong T-Consistency, Verifiable Secret Sharing, Verifiable Polynomial Differential

## I. Introduction

In 1979, Shamir introduced the threshold secret sharing scheme as a solution for safeguarding cryptographic keys. In [1], his scheme triggered the consideration of how a secret key could be shared with multiple participants. Shamir's SSS has been used as the foundation for wide research in the computer security field. Thus, an overview of his important scheme is provided, followed by a simple example of his scheme.

Basic secret sharing schemes assume that the dealer who divides the secret and distributes the shares of the secret to participants is a mutually trusted party. In 1985, Chor et al. [2] extended the original secret sharing approach and presented the concept of verifiable secret sharing. The property of verifiability enables participants to verify that their shares are consistent (a set of $n$ shares is $t$-consistent if any subset containing $t$ shares defines the same secret $S$).

There are two types of verifiable secret sharing schemes: interactive VSS and non-interactive VSS. In interactive VSSS, participants can only prove and verify their shares are $t$-consistent by making contact with the other participants online (who are selected in the secret reconstruction phase). In non-interactive VSSS, the participants can verify their shares are $t$-consistent without talking to each other or with the dealer. Both types allow the validity of the secret share to be verified without the share being revealed. Both types of VSSS are based on the property that says: if the sum of two polynomials is of degree at most $t-1$, then either both are of degree at most $t-1$ or both are of degree greater than $t-1$.

In [3], Benaloh proposed an interactive VSSS. In Benaloh's scheme, the dealer chooses a secret polynomial $f(x)$, and then generates and distributes shares $f(i)$, $\forall i=1,...n$ for each corresponding participant $P_i$, $\forall i=1,...n$ as in Shamir's scheme. The dealer chooses $k$ (say 100) polynomials $f_j(x)$, $\forall j=1,...k$ of degree $t-1$ exactly. For each polynomial $f_j(x)$, the dealer follows the same step to generate and distribute $k$ sub-shares $f_j(i)$ for each participant $P_i$. In the share verification algorithm, only the $t$ participants who want to reconstruct the secret ask the dealer to reveal $k/2$ polynomials to ensure that all these $k/2$ revealed polynomials are of degree $t-1$ exactly. With high probability, they can ensure that the remaining unrevealed polynomials $k/2$ are of degree $t-1$ exactly. The dealer reveals the polynomial $F(x)=f(x)+f_j(x)$, $\forall j=1,...k/2$ which is the sum of the secret polynomial $f(x)$ and the remaining unrevealed polynomials $f_j(x)$. Each participant $P_i$ can verify that his or her share is the real share of the secret by finding the sum of their secret share and the remaining unrevealed sub-shares as $F(i)=f(i)+f_j(i)$, $\forall j=1,...k/2$ based on the homomorphism of the polynomials (Definition 1). In the secret reconstruction algorithm, the $t$ participants can pool their secret shares and use the Lagrange interpolation polynomial (Equation 2.1) to find the secret.

**Definition 1**: *(homomorphism of polynomials) [3]: Suppose* $f_i(x)$, $\forall i=1,...,n$ *are polynomials, and* $F(x)=\sum_i^n f_i(x)$ *is the additive sum of these polynomials. Then for each integer* $x_j$, $\forall j=1,...,n$ *the*

 

*additive sum of shares* $\sum_{i=1}^{n} f_i(x_j) = F(x_j)$ , $\forall j = 1,...,n$ *of polynomials* $f_i(x)$, $\forall i = 1,...,n$ *is the share of additive sum of polynomials* $F(x) = \sum_{i=1}^{n} f_i(x)$.

According to the security property, the verifiable secret sharing schemes can be classified into two types: the computationally secure schemes [3, 4] and the unconditionally secure schemes [5, 6]. For example, Feldman's VSSS [4] is a computationally secure scheme based on the difficulty of solving the discrete logarithm problem, Nikova's VSSS [5] and Pederson's VSSS [6] are unconditionally secure schemes. Furthermore, there are VSSS based on some computational assumptions.

This paper first provides an overview of Shamir's scheme, followed by an overview of Pedersen's approach to remove the assumption in Feldman's VSSS and propose a VSSS which is information-theoretically secure. The information-theoretically secure strong $(t, n)$-VSS of [7] is then presented. Our proposed scheme the polynomial differential-based information-theoretically secure scheme that satisfies both definitions of $t$-consistency and strong $t$-consistency is then proposed.

## II. Previous Study

In this section we have to mention, in general, the important threshold schemes in secret sharing is Shamir's scheme [1] , the second part will be about Pedersen's (n, t, n) -SSS, and finally in this section will be overview about strong (t, n) -VSS [8].

### A. Shamir's SSS

In 1979, Shamir introduced the threshold secret sharing scheme as a solution for safeguarding cryptographic keys. In [1], his scheme triggered the consideration of how a secret key could be shared with multiple participants. Shamir's SSS has been used as the foundation for wide research in the computer security field. Thus, an overview of his important scheme is provided, followed by a simple example of his scheme.

Let $X = \{P_1, P_2,.., P_n\}$ the set of $n$ participants, and a dealer D. Let $p$ be a large prime number, and a finite field $Z_p$. The scheme consists of two algorithms, namely, the share generation algorithm and the secret reconstruction algorithm, which are explained as follows:

1. In the share generation algorithm, D does the following:

(a) Randomly picks a polynomial $f(x)$ of degree $t-1$ , $f(x) = a_0 + a_1 x + .. + a_{t-1} x^{t-1}$ , in which the secret $S = f(0) = a_0$ and all coefficients $a_1,..., a_{t-1}$ are in $Z_p$.

(b) Computes the shares: $s_1 = f(1),..., s_n = f(n)$.

(c) Outputs a list of $n$ shares $s_i$ , $\forall i = 1,..., n$ and privately distributes each share to the corresponding participants $P_i, \forall i = 1,.., n$.

2. Secret reconstruction algorithm – In Shamir's $(t, n)$-SSS the reconstruction of a secret is based on the Lagrange interpolating polynomial (see Equation (1)).

Let $Y = \{1,2,...,n\}$ a finite set of positive integers, and $A = \{x_1,..., x_t\} \subseteq Y$ , where $x_i \neq x_j$ , $\forall i \neq j$ , $\forall i, j = 1,..., t$.

Given the distinction of $t$ shares $s_{x_1},..., s_{x_t}$ , we can reconstruct the secret *S* as $t$ participants who work together, and then apply the Lagrange interpolating polynomial such as:

$$f(x) = \Sigma_{i=1}^{t} (s_{x_i} \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{x - x_j}{x_i - x_j}) \quad (1)$$

Then the secret $S = f(0) = \Sigma_{i=1}^{t} (s_{x_i} \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{x_j}{x_j - x_i})$

The above scheme satisfies the basic security requirements of the SSS as follows:

- With knowledge of any *t* or more than *t* shares, it can reconstruct the secret *S*.
- With knowledge of any fewer than *t* shares, the secret *S* cannot be reconstructed.

### B. Pedersen's Information-Theoretically Secure VSSS

In 1992, Pedersen [6] proposed a non-interactive and information-theoretically secure VSSS based on Feldman's VSSS. In Shamir's $(t, n)$-SSS [1], a dealer is a trusted third party who generates and distributes shares to participants by using polynomial $f(x)$.

**Definition 2** *$t$-consistency of shares [6]: A set of $n$ shares is $t$-consistent if any subset containing $t$ shares defines the same secret.*

Shamir's scheme is information-theoretically secure since the scheme satisfies the security requirement without making any computational assumptions. In Shamir's scheme, $t$ participants (and more) can reconstruct the secret S, and fewer than *t* participants know nothing about the secret *S* and cannot reconstruct it. In [3], Benaloh observed that a share in Shamir's $(t, n)$-SSS is $t$-consistent if and only if the interpolation of shares yields a polynomial of degree at most $t-1$.

In Feldman's VSS scheme [4], the committed values are publicly known and the privacy of secret *S* depends on the difficulty of solving the discrete logarithm problem. In other words, Feldman's scheme is computationally secure (i.e., the scheme is based on some computational assumptions). Pedersen [6] proposed a non-interactive and information-theoretically secure VSSS based on Feldman's VSSS [10].

 

To illustrate, let $p$ and $q$ be two large primes such that $q|(p-1)$, and $g,h \in Z_p$ are two elements of order *q*. Let $X=\{P_1,P_2,...,P_n\}$ be a set of $n$ participants. Pedersen's scheme can be described as follows:

1. In the share generation algorithm, the dealer D:

(a) Randomly picks a polynomial $f(x)=a_0+a_1x+...+a_{t-1}x^{t-1}$ of degree at most $t-1$ in which the secret $S=f(0)=a_0$ and all coefficients $a_1,a_2,...,a_{t-1}$ are in $Z_p$.

(b) Picks $b_0,b_1,...,b_{t-1}$ at random. Let $k(x)=b_0+b_1x+...+b_{t-1}x^{t-1}$.

(c) Computes shares $(s_i,t_i)$ $\forall i=1,..,n$ and each coefficient's commitment of the added sum of polynomials of $f(x)$ and $k(x)$ as follows: $(s_i,t_i)=(f(i),k(i))$.

(d) Computes $c_j=g^{a_j}h^{b_j} \pmod p$, $\forall j=0,1...,t-1.$

(e) Outputs a list of $n$ shares $(s_i,t_i)$ and privately distributes each share to the corresponding participant $P_i$. D also broadcasts $c_j$ for all participants.

2. In the share verification algorithm, each participant $P_i$ who has received the share $(s_i,t_i)$ and all broadcasted information $c_j$, can verify that the share defines a secret *S* by testing:

$$g^{s_i}h^{t_i} \pmod p = \prod_{j=0}^{t-1} c_j^{i^j} \pmod p$$

3. As in Shamir's scheme, the secret reconstruction algorithm operates such that any *t* participants and more than *t* participants can reconstruct the secret and fewer than *t* participants cannot reconstruct it.

In Pedersen's scheme, the value $g^{a_0}$ is not made publicly known, that is, the secret $S=a_0$ is embedded in the commitment $g^{a_0+ub_0}$, where $u=log_g h$. Thus, no information is directly known about the secret *S*. Even if an attacker with unlimited computing power can solve $u=log_g h$, the attacker still gets no information about the secret *S*. In Pedersen's scheme, if an attacker can guess the commitment $b_0$, then the security in his scheme will be computational security. The scheme proposed in this paper (Section 3) fixes this problem by using a secret verification polynomial which acts as a random polynomial in Pedersen's VSSS and in a strong $(t,n)$-VSSS (Section 2.2). Thus, the coefficients in that verification polynomial are not publicly known.

The next section provides an overview of the strong $(t,n)$-VSSS [24] which is an important scheme that satisfies the definition of strong *t*-consistency (see Definition 3 in the next section); but, at the same time, the verification polynomial used in this scheme is publicly known. Its role is only to ensure the used polynomial $f(x)$ is exactly of degree $t-1$, and then satisfy the security requirements in *t*-consistency.

### *C. Information-Theoretically Secure Strong (t, n)-VSSS*

Changlu, Harn and Dingfeng [24] observed in Pedersen's SS [6] that the *t*-consistency of the shares does not guarantee that the shares satisfy the security requirement. They assumed that the used polynomial $f(x)$ can be of degree $t-2$ to generate shares. The shares are then *t*-consistent, but the threshold of shares is $t-1$. These shares violate the security requirement of a $(t,n)$-SSS, since any set of $t-1$ shares can reconstruct the secret. They proposed the definition of strong *t*-consistency to fix this security problem. Their scheme is called strong-$(t,n)$-VSSS.

**Definition 3** *Strong t-consistency [7]: A set of n shares is strongly t-consistent if (a) any subset containing t or more than t shares defines the same secret; and (b) any* $t-1$ *or fewer than* $t-1$ *shares cannot define the same secret.*

Changlu, Harn and Dingfeng [7] noted that if the shares in Shamir's $(t,n)$-SSS are generated by a polynomial of degree $t-1$ exactly, these shares are strongly *t*-consistent and satisfy the security requirements of a $(t,n)$-SSS. They observed that Pedersen's scheme satisfied *t*-consistency without guaranteeing the security requirements of a secret sharing scheme. On the other hand, the polynomial used in Pedersen's scheme which shares the secret is not guaranteed to be of degree $t-1$ exactly. The scheme proposed by Changlu, Harn and Dingfeng is illustrated below.

Let $p$, $q$, $g$, $h$ be the same as the parameters in Pedersen's scheme; a dealer D will divide a secret $S \in Z_p$. The scheme proposed by Changlu, Harn and Dingfeng is described as follows:

1. In the share generation algorithm, D:

(a) Randomly picks a secret polynomial $f(x)=a_0+a_1x+a_2x^2+...+a_{t-1}x^{t-1}$ of degree at most $t-1$, and all coefficients $a_0,a_1,..a_{t-1}$ are in $Z_p$.

(b) Randomly picks a public polynomial $f'(x) = a_0' + a_1'x + a_2'x^2 + ... + a_{t-1}'x^{t-1}$ where $a_0', a_1', a_2', ..., a_{t-1}' \in Z_p$. Let $F(x) = f(x) + f'(x)$ in which the secret $S = a_0 + a_0' = F(0)$.

(c) Computes $F_i = a_i + a_i', \forall i = 0,..,t-1$.

(d) Randomly picks $b_0, b_1, ..., b_{t-1}$, and lets $k(x) = b_0 + b_1x + ... + b_{t-1}x^{t-1}$.

(e) Computes the initial share $(s_i, t_i)$ as follows: $(s_i, t_i) = (f(i), k(i))$, $\forall i = 1,...,n$.

(f) Computes the value $c_j = g^{F_j} h^{b_j} \mod p$, $\forall j = 0,...,t-1$.

(g) Outputs a list of $n$ initial shares $(s_i, t_i)$ and privately distributes each share to the corresponding participant $P_i$. D also broadcasts $c_j$ for all participants.

(h) Each participant $P_i$ computes the real share $S_i = s_i + f'(i)$.

2. In the share verification algorithm, each participant $P_i$, $\forall i = 0,...,n$ who has received the initial share $(s_i, t_i)$ and all the broadcasted information $c_j$, $\forall j = 0,...,t-1$ can verify that his share defines the secret by testing:

$$g^{S_i} h^{t_i} \text{ (mod p)} = \prod_{j=0}^{t-1} c_j^{i^j} \text{ (mod p)}$$

3. The secret reconstruction algorithm is the same as in Shamir's scheme: any $t$ participants and more than $t$ participants can reconstruct the secret and fewer than $t$ participants cannot reconstruct it.

It is observed that the difference between the strong $(t, n)$-VSSS and Pedersen's scheme is that shares in the strong $(t, n)$-VSS are combined in two polynomials. The first one is a secret polynomial, and the second one is a public polynomial. The public polynomial acts to ensure that the secret polynomial is of degree $t-1$ exactly so that the strong $(t, n)$-VSS satisfies the security requirement of $t$-consistency.

It is noted that the public polynomial is not secure in [7]. Thus, the scheme proposed by Changlu, Harn and Dingfeng still has the same security issue as in Pedersen's scheme. In both schemes, the broadcasted information $c_j$ still contains information about the secret. The strength in both schemes depends on the difficulty of solving the discreet logarithm problem.

In Pedersen's VSSS, the secret is embedded in commitments, and the polynomial used is at most of degree $t-1$. In the verification stage of the strong $(t, n)$-VSSS, that polynomial is public, but not secure. It only satisfies the security requirement in the definition of $t$-consistency.

## III. Polynomial Differential-Based Information-Theoretically Secure VSSS

In Pedersen's VSSS, the secret is embedded in commitments, and the polynomial used is at most of degree $t-1$. In the verification stage of the strong $(t, n)$-VSSS, that polynomial is public, but not secure. It only satisfies the security requirement in the definition of $t$-consistency.

The scheme proposed in this section satisfies the security requirement of strong $t$-consistency (see Theorem (3.1)) and it considers the security on the verification polynomial based on Pedersen's VSSS [6], where the secret is added into a sum of two secret values. In the proposed scheme, the dealer creates a polynomial $F(x)$ of degree $2t-1$, where $t$ is the threshold. Then the dealer calculates the $t^{th}$ derivative of the polynomial. This derivative polynomial serves the purpose of verification. The verification polynomial is also a secret as it is the polynomial used which shares the secret $S$. Finally, the dealer applies Shamir's SSS to generate and distribute the verification shares.

The scheme proposed in this section shows that the verification polynomial has two roles. The first one is to ensure that the shares satisfy a strong $t$-consistency, and the second role is to verify that the shares are $t$-consistent. The discussion in this section shows that the proposed scheme is more secure than Pedersen's VSSS [6] and strong $(t, n)$-VSSS [7].

Let $p$, $q$, $g$, $h$ be the same parameters as in Pedersen's VSSS; the dealer D will divide the secret $S \in Z_p$ for $n$ participants. The proposed scheme is described as follows:

1. In the share generation algorithm, D:

(a) Creates polynomial $F(x) = f_1(x) + f_2(x) = \Sigma_{k=0}^{t-1} a_k x^k + \Sigma_{k=t}^{2t-1} a_k x^k$ of degree $2t-1$ (i.e., $f_1(x) = a_0 + a_1x + ... + a_{t-1}x^{t-1}$, and $f_2(x) = a_t x^t + ... + a_{2t-1}x^{2t-1}$), and the secret will be $S = F(0) = f_1(0) = a_0$, such that all coefficients of the polynomial $F(x)$ are in $\in Z_p$

(b) Derives $F(x)$ $t^{th}$ times, in order to calculate the verification polynomial $f_2^{(t)}(x)$, where $f_2^{(t)}(x) = \Sigma_{k=0}^{t-1} \frac{(t+k)!}{k!} a_{t+k}^k$; this derivative polynomial will be of a degree $t-1$.

(c) Makes $R(x) = f_1(x) + f_2^t(x)$ ; that is, $R(x) = R_0 + ... + R_{t-1}x^{t-1}$ , and lets $R_j = a_j + \frac{(t+j)!}{j!}a_{j+t}$, $\forall j = 0,..,t-1.$

(d) Randomly picks $b_0, b_1, ..., b_{t-1} \in Z_p$ , lets $k(x) = b_0 + b_1 x + ... + b_{t-1}x^{t-1}$ , and computes $t_i = k(i)$, $\forall i = 0,..,n$.

(e) Calculates the share as $R(i) = s_i$, $\forall i = 0,..,n$.

(f) Calculates the verification share as $f_2^t(i) = v_i$ , $\forall i = 0,..,n$.

(g) Computes the value $c_j = g^{R_j} h^{b_j} \pmod{p}$, $\forall j = 0,..t-1$.

(h) Outputs a list of $n$ shares $(s_i, v_i, t_i)$ and privately distributes each share to the corresponding participant $P_i$, $\forall i = 1,..., n$. D also broadcasts $c_j$, $\forall j = 1,..., t-1$ for all participants.

It is noted that the secret $a_0$ is added to another secret value $a_t$ as follows: $R_0 = a_0 + t!a_t$ , and then the broadcasted $c_j$ still conveys nothing about the secret.

2. In the share verification algorithm, each participant $P_i$ who has received the share $(s_i, v_i, t_i)$ and all the broadcasted information $c_j$, can verify that the share is a correct share by testing :

$$g^{R_i} h^{t_i} \pmod{p} = \prod_{j=0}^{t-1} c_j^{i^j} \pmod{p}$$

The security strength of the proposed scheme lies in the fact that the shares and all the broadcasted information convey no information about the secret $S = F(0) = a_0$.

In the share verification algorithm, the secret is perfectly secure. Thus, the algorithm still gives no information about the secret $S = F(0) = a_0$. The shares and all the broadcasted information provide no information about the secret, even if the attacker succeeds in resolving the discrete logarithm problem of $c_0 = g^{R_0} h^{b_0}$ because the secret $a_0$ is added to a secret value ( $t!a_t$ ). This means that the proposed scheme is information-theoretically secure.

3. To illustrate the secret reconstruction algorithm, suppose $\{A = P_1, ..., P_t\}$ is the set of $t$ participants who want reconstruct the secret *S*. The participants do the following:

(a) Pool their verification shares $v_i$ , and then use the Lagrange interpolation (Section 2.1) in order to reconstruct both polynomials $f_2^{(t)}(x)$, $f_2(x)$, and then verify that these polynomials have degree $t-1$.

(b) Calculate the real secret shares $S_i = s_i - f_2(i)$, $\forall i = 1,..., t$ , and then use the Lagrange interpolation again to find the secret *S*.

Next, Theorem (3.1) demonstrates that the proposed scheme satisfies the basic requirements of a secret sharing scheme as in Shamir's scheme:

- With knowledge of any *t* or more than *t* shares, the secret *S* can be reconstructed.
- With knowledge of less than *t* shares, the secret *S* cannot be reconstructed.

This means that the proposed scheme has strong *t*-consistency.

**Theorem 3.1**: *If the verification shares* $v_i$, $i = 1,..., t$ *construct the verification polynomial* $f_2^{(t)}(x)$ *of degree* $t-1$ *exactly, and the real secret shares* $S_i$, $i = 1,..., t$ *also construct the secret polynomial* $F(x)$ *of degree* $2t-1$ *exactly, then the proposed scheme has strong t-consistency.*

**Proof**: If the secret polynomial $F(x)$ is of degree $2t-1$ exactly, then $2t$ shares and more than $2t$ shares is needed in order to reconstruct it (the secret reconstruction algorithm needs *t* participants working together and pooling their shares $(s_i, v_i)$ to reconstruct $F(x)$ ). Before the secret reconstruction algorithm is completed, and when the participants successfully complete the share verification algorithm by using *t* verification shares $v_i$, then they can guarantee that the verification polynomial $f_2^{(t)}(x)$ has a degree of $t-1$ exactly. Then the secret is still secure, and the shares are satisfying the definition of strong *t*-consistency ((a) any subset containing *t* or more than *t* shares defines the same secret; and (b) any $t-1$ or fewer than $t-1$ shares cannot define the same secret).

## VI. Security Analysis

In Feldman's VSSS, the committed values are publicly known and the privacy of a secret *S* depends on the difficulty of solving the discrete logarithm problem. In other words, Feldman's scheme is computationally secure. Pedersen proposed a non-interactive and information-theoretically secure VSSS based on Feldman's scheme. In Pedersen's scheme, the value $g^{a_0}$ is not made publicly known; that is, the secret $S = a_0$ is embedded in the commitment $c_0 = g^{a_0} h^{b_0}$ . Thus, there is no

information available about the secret *S*. Only if an attacker with unlimited computing power succeeds in solving the discrete logarithm problem, can he obtain some information about the secret *S* (if an attacker can only guess the commitment $b_0$, then Pedersen's scheme will be computationally secure).

In a strong $(t,n)$-VSSS [7], the public polynomial is not secure. The scheme still has the same security as in Pedersen's scheme. In both schemes, the broadcasted information $c_0$ still contains information about the secret. The strength in both schemes depends on the difficulty of solving the discreet logarithm problem.

The proposed scheme used the polynomial $F(x)$ of degree $2t-1$ which is divided into two secret polynomials $f_1(x)$ and $f_2(x)$. Each one needs $t$ shares to construct the secret, as the secret is embedded in the derivative polynomials $f_2^{(t)}(x)$ such that the secret is the sum of the free coefficients in $f_1(x)$ and $f_2^{(t)}(x)$. The difference between the proposed scheme and the last two schemes (Pederson and strong $(t,n)$ schemes) is that the share verification algorithm still conveys no information about the secret $a_0$. This is because the broadcasted information $c_0 = g^{a_0+!ta_t} h^{b_0} = g^{a_0+!ta_t+ub_0}$ adds the secret $a_0$ to another secret value $a_t$, where $u = \log_g h$. There is no information about the secret $a_0$ even if an attacker succeeds in finding the commitment $b_0$ and solves the discrete logarithm problem of $u = \log_g h$. The proposed scheme guarantees that it is information-theoretically secure. It is more secure than both the Pederson and strong $(t,n)$ schemes.

## V. Conclusion

The contribution in this paper is the development of a polynomial differential-based information-theoretically secure scheme that satisfies both definitions of $t$-consistency and strong $t$-consistency. As discussed, when the participants successfully complete the share verification algorithm, the participants still know no information about the secret *S*. The proposed scheme succeeds in eliminating the problem of an attacker resolving the difficulty of the discrete logarithm problem (the broadcasted information $c_j$ tells nothing about the secret, unlike Pederson's scheme and the strong $(t,n)$ scheme when the commitment value is not secure). In addition, this paper considered whether the proposed scheme satisfies the basic requirements of a secret sharing scheme. It was concluded that the proposed scheme is more secure than Pederson's scheme and the strong $(t,n)$ scheme, and it is information-theoretically secure.

## References


[1] A. Shamir. How to share a secret. Communications of the ACM, 1979.

[2] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, 21–23 October, Oregon, Portland, IEEE Computer Society, 1985, pp. 383–395.

[3] J.C. Benaloh, 1987. Secret sharing homomorphism: keeping shares of a secret. In: Advances in Cryptology, Proceedings of the Crypto'86, vol. 263, 11–15 August, Santa Barbara, California, USA, LNCS. Springer-Verlag, Berlin, pp. 251–260.

[4] P. Feldman, 1987. A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th IEEE Symposium on Foundations of Computer Science, 27–29 October. IEEE Computer Society, Los Angeles, California, pp. 427–437.

[5] V. Nikov, Nikova, S., 2005. On a Relation Between Verifiable Secret Sharing Schemes and a Class of Error-Correcting Schemes, Cryptology e-print archive 2003/210.

[6] T.P. Pedersen, 1992. Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances in Cryptology-CRYPTO'91, LNCS, vol. 576. Springer-Verlag, Berlin, pp. 129–140.

[7] C. Lin, L. Harn, D. Ye: Information-theoretically Secure Strong Verifiable Secret Sharing. SECRYPT 2009: 233-238.


**Author's Profiles**


**Qassim Al Mahmoud** received his B.S. degree in Mathematics and Computer Science from University of Baghdad in Iraq, 2002. He received his M.S. degree in Information Technology from University of Arab Academic for accounting and Information Technology in Jordan, 2007. Now he is a Ph.D. candidate in the Information Security, Faculty of Mathematics and Computer Science, University of Bucharest, Romania.



**How to cite this paper:** Qassim Al Mahmoud,"Polynomial Differential-Based Information-Theoretically Secure Verifiable Secret Sharing", International Journal of Information Technology and Computer Science(IJITCS), vol.6, no.12, pp.18-23, 2014. DOI: 10.5815/ijitcs.2014.12.03


 