Sensors & Transducers, Vol. 209, Issue 2, February 2017, pp. 65-73

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2017
http://www.sensorsportal.com

An Ideal Multi-Secret Sharing Scheme with Verification

[1, *] Arup Kumar CHATTOPADHYAY, [2] Amitava NAG and [3] Koushik MAJUMDER

[1] Department of Computer Science and Engineering, Academy of Technology, Hooghly, 712121, India
[2] Department of Information Technology, Academy of Technology, Hooghly, 712121, India
[3] Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, 700064, India
[*] Tel.: 9339770502
[*] E-mail: ardent.arup@gmail.com

Received: 23 January 2017 /Accepted: 23 February 2017 /Published: 28 February 2017

Abstract: The (t,n) - threshold secret sharing scheme proposed by Shamir (1979) encodes the secret key into n shares and those shares are distributed among the participants. Each share incorporates an *x*-value uniquely designated for every single participant. If the shares are distributed on insecure (or public) channels, malicious users may acquire a few (or all) shares. If the number of shares obtained by the malicious users is t or more ($\leq n$), then malicious users can reconstruct the secret. Chattopadhyay, *et al.* (2016) proposed a scheme for the distribution of shares with RSA to encrypt the *x*-values at insecure channel along with verification, such that the secrecy of shares and secret both are maintained. A verifiable $(t,k,n)-$ threshold multi-secret sharing scheme is proposed in this paper which extends the work proposed by Chattopadhyay, *et al.* (2016). The proposed scheme is ideal as the sizes of the shares are of same as that of the secrets.

Keywords: Secret sharing, Multi-secret sharing, Verifiable, Hash function, RSA.

1. Introduction

A secret sharing schemes (SSS) are defined to share a secret among a group of participants and protect the secret from disloyal access by unauthorized groups. In a (t,n) - threshold secret sharing scheme, a secret S is encoded into n shadows or shares and distributed among n participants or players (a Dealer is responsible for construction and distribution of the shares). If any t or more $(\leq n)$ shares are submitted by participants, then the full secret S can be reconstructed (reconstruction can be also done by a special entity called Combiner). Otherwise, no less than t shares can reconstruct the secret or expose any information about the secret. So, any group with $<t$ members constitutes a disloyal group.

Secret sharing was first introduced independently by two authors - Shamir and Blakley in the year of 1979. Shamir's scheme [3] was based on Lagrange's interpolation and Blakley's scheme [7] was based on hyperplane geometry. Mignotte proposed secret sharing scheme [17] based on Chinese Reminder Theorem (CRT) and it was improved by Asmuth-Blooms [4].

The scheme proposed by Shamir [3] is extended by many researchers in different scenarios. Thien and Lin proposed secret image sharing (SIS) scheme in [5] based on Shamir's scheme. If the shares generated by Shamir's scheme, are distributed on insecure channels,

the confidentiality of shares becomes volatile and the shares can be misused by malicious users. Hence, Zhao, *et al.* has proposed a method in [21] to ensure the confidentiality of the shares on insecure channels. The secure key distribution method used in [21] is extended for secure distribution of shares of medical images by Ulutas, *et al.* in [18] and the authors also used the Shamir's framework with better authenticity and confidentiality properties.

During the reconstruction of secret phase, when the shareholders present their shares, dishonest shareholder(s) called cheater(s) can present faked share(s) and thus deceive the other honest shareholders as they obtain a faked secret as result. So, cheater detection and identification are essential properties of an efficient secret sharing scheme. Harn and Lin [11] defined a method for cheater detection and identification for Shamir's secret sharing scheme. Authors in [11] assume a situation where more than t shares are presented for reconstruction of the secret and the redundant shares are used to identify the cheaters. A variety of secret sharing schemes were proposed that can verify whether the shares received by shareholders are consistent under the condition the secrecy of shares and secret both are maintained. Harn, *et al.* [12] proposed a verifiable secret sharing scheme based on the CRT and extension of Asmuth-Bloom's scheme [4]. Another efficient scheme [22] based on Asmuth-Blooms scheme is proposed by Liu, *et al.* Liu & Chang had proposed an integrable mechanism for verification in [23] with generalized Chinese Reminder Theorem, Shamir's Secret sharing and Asmuth-Bloom's secret sharing and it improvise the verification method proposed by Harn, *et al.* in [12] by using one way hash function.

In multi-secret sharing (MSS) schemes, several secrets can be shared during a single secret sharing process. In 2004, Yang, Chang and Hwang (YCH scheme in [6]) proposed an efficient MSS, which is based on the two-variable one-way function. But the scheme doesn't have the property of verification. Based on YCH scheme, a number of MSS schemes are proposed to realize the property of verification and also do not need a security channel for delivering shares to the participants. Shao and Cao (SC scheme in [10]) proposed an efficient verifiable multi-secret sharing (in 2005) based on YCH scheme, but in the scheme as the shadows are chosen by the dealer, even if the dealer is honest, the system also needs a secure channel between the dealer and the participants so that the dealer can distribute the shadows to the participants safely. In 2007, Zhao, *et al.* presented a new practical verifiable MSS scheme (ZZZ scheme in [8]) based on YCH scheme and the intractability of the discrete logarithm [19] which ensures the verifiability of shares and secrets both and do not require a secret channel for distribution of shares. In the same year, Dehkordi and Mashhadi presented another MSS scheme [15] based on based on- YCH scheme, the intractability of Discrete Logarithm and RSA cryptosystem. The scheme needs no secure channel and at the same time verifiable property is more efficient than defined in [10]. The same authors, in 2008 presented another MSS scheme [16] based on YCH and homogeneous linear recursion, which is performance wise better than SC and ZZZ schemes. Shao (2014) presented another MSS scheme [9] based on Shamir's scheme and a hash function. The scheme, in [9] is computationally efficient and holds all the properties of MSS.

In our proposed multi secret scheme we ensure that it holds cheating prevention – shares are secure even distributed on public channels and also the shares and the secrets both are verifiable. Moreover, the proposed scheme is ideal as the shares are of the same size as that of the secret. The rest of the of the paper is arranged as follows – in Section 2 the important entities of threshold secret sharing are briefed; we have discussed about the related schemes and algorithms in Section 3; our proposed scheme is presented in Section 4; in Section 5 we have discussed about the performance of our scheme and Section 6 concludes the paper.

2. Preliminaries

The important entities used in a threshold secret sharing scheme are briefed as follow:

Secret: A secret S is the confidential information need to be secured from unauthorized users or unauthorized groups.

Shares: The secret S is encoded into n shares or shadows, say $s_1, s_2, ..., s_n$, such that none of them individually reveals any information about the secret.

Dealer: Dealer or distributor (*D*) is the data owner or trusted-third party, mainly responsible to encode the secret into n shares and to distribute them to the participants.

Participants: Participants or players are represented as $P_1, P_2, ..., P_n$ and they are the users seeking for the secrets.

Combiner: A combiner (*C*), one of the participants or trusted-third party, mainly responsible to decode the secret if threshold number (*t*) or more shares are obtained from the participants.

In a (t, n)-threshold secret sharing scheme, the *D* generates n *shares* and distributes it among n *participants*. If any *t* or more *participants* submit their *shares* to the *C*, then the secret can be retrieved in full. If any less than *t shares* are submitted then no part of the secret will be revealed.

3. Related Study

3.1. Review of Shamir's Secret Sharing Scheme (1979)

Shamir proposed a (t, n) – threshold secret sharing scheme [3] based on Lagrange's interpolation. For

given t points (x_i, y_i), where $i = 1, 2, ..., t$ in the *2D*-plane, the Lagrange's interpolation polynomial $f^t(x)$ can be constructed using:

$$f^t(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j}.$$

Consider the secret is S and the dealer has to generate n shares as - $s_1, s_2, ..., s_n$. It allows t or more $(\leq n)$ shares to reconstruct the secret. The solution requires a random $(t-1)^{\text{th}}$ degree polynomial:

$$F(x) = \left(S + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}\right) \bmod p,$$

in which p is the large prime number and the coefficient of polynomial $a_1, a_2, ..., a_{t-1}$ are randomly selected within the range $[0, (p-1)]$. Dealer computes the shares as follows:

$$s_1 = (1, F(1)), s_2 = (2, F(2)), ..., s_n = (n, F(n)).$$

If t or more $(\leq n)$ shares are obtained then the polynomial $F(x)$ can be regenerated by Lagrange interpolation as:

$$F(x) = \sum_{i=1}^{t} S_i \prod_{j=1, j\neq i}^{t} \frac{x - j}{i - j}$$

The secret will retrieved as $S = F(0)$.

3.2. Threats in Distribution of Shares in Shamir's Secret Sharing

Each share generated by Shamir's scheme, is distributed as pair of two integers $(x_i, F(x_i))$, where $x_i \neq 0$. But in this scheme the *x*-values are easy to predict. So *x*-values can be considered as random numbers. But still, if the shares are distributed on insecure channels, any knowledge of *t* or more $(\leq n)$ shares were gathered by malicious users are enough to reconstruct the polynomial - $F(x)$. The value of the polynomial at position $x_i = 0$ is the secret $S = F(x_i = 0)$.

For example, we consider $(3,10)$ – threshold secret sharing $(n = 10, t = 3)$ and a 2nd order polynomial as:

$$F(x) = \left(9 + 13x + 5x^2\right) \bmod 37$$

All the coefficients are assumed within [0-36], secret $S = 9$ and $p = 37$. The polynomial can be reconstructed (as shown in Fig. 1) if 3 or more shares are known. The secret will be recovered where the polynomial intersects *Y*-axis that is at point $(0, F(0))$. In our example secret is $S = F(0) = 9$.

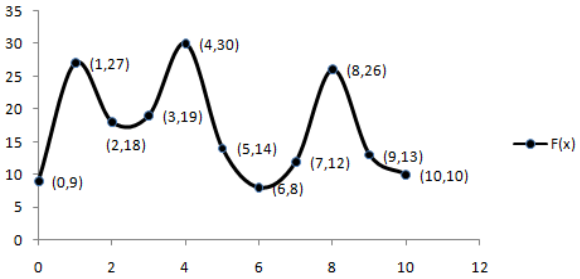

Fig. 1. Example of $(3,10)$-threshold SSS.

3.3. Review of the YCH Scheme (2004)

C.-C. Yang, T.-Y. Chang and M.-S. Hwang proposed a (t,n)-threshold multi-secret sharing scheme [6] which is based on Shamir's secret sharing. It assumes a two-variable, one-way function $f(r,s)$ and k secrets - $S_1, S_2, ..., S_n$.

Initialization Phase:

The dealer (*D*) randomly chooses *n* secret shadows - $s_1, s_2, ..., s_n$ and delivers them to the participants $P_1, P_2, ..., P_n$ secretly. Then, *D* chooses a value r randomly and computes $f(r, s_i)$ for $i = 1, 2, ..., n$.

Construction Phase:

if $(k \leq t)$

1. Choose a polynomial of degree $(t-1)$

$$g(x) = S_1 + S_2 x^2 + ... + S_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + ... + a_{t-k} x^{t-1} \bmod q,$$

where q is the large prime and $0 < S_1, S_2, ..., S_k, a_1, a_2, ..., a_{t-k} < q$.

2. *D* computes $y_i = g\left(f(r, s_i)\right) \bmod q$ for $i = 1, 2, ..., n$.

3. *D* publishes $(r, y_1, y_2, ..., y_n)$.

if $(k > t)$

1. Choose a polynomial of degree $(k-1)$

$$g(x) = S_1 + S_2 x + ... + S_k x^{k-1} \bmod q;,$$

where q is a large prime and $0 < S_1, S_2, ..., S_k$.

2. *D* computes $y_i = g\left(f(r, s_i)\right) \bmod q$ for $i = 1, 2, ..., n$. *D* also computes $g(i) \bmod q$ for $i = 1, 2, ..., k - t$.

3. *D* publishes

$$\left(r, y_1, y_2, ..., y_n, g(1), g(2), ..., g(k-t)\right).$$

Recovery Phase:

Let, any t participants submit their shares $f(r, s_i) \bmod q$ for $i = 1, 2, ..., t$. Then the polynomial $g(x)$ can be uniquely determined as follows:

$$if\left(k \le t\right)$$

$$g(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \bmod q$$

$$= S_1 + S_2 x + ... + S_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + ...$$

$$+ a_{t-k} x^{t-1} \bmod q;$$

$$if\left(k > t\right)$$

$$g(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \bmod q$$

$$+ \sum_{i=1}^{k-t} g(i) \prod_{j=1, j\neq i}^{k-t} \frac{x - j}{i - j} \bmod q$$

$$= S_1 + S_2 x + ... + S_k x^{k-1} \bmod q;$$

But, as in this scheme shadows are chosen by the dealer, even if the dealer is honest, the system also needs a secure channel between the dealer and the participants so that the dealer can distribute the shadows to the participants safely. Hence, it increases the cost of distribution. In 2009, Zhao, *et al.* proposed a scheme [21] which do not require secret channel (for Shamir's secret sharing). In the next subsection, we review only the initialization and sharing phase of it.

3.4. Scheme Proposed by Zhao, *et al.* (2009)

The intercommunication between dealer and participants to distribute the shares on an insecure channel was made secure (confidentiality of the shares is not lost) by the cheating proof secret image sharing scheme [21] proposed by Zhao, *et al*. It ensures the confidentiality of *x*-values are protected as the *x*-values are calculated independently by both dealer and participant before distribution begins. Even if intruders able to gather the information about any t or more $(\le n)$ shares (only the values of $F(x)$), they must also have to accrue the *x*-values to apply Lagrange's interpolation to recover the secret. Thus, even an insecure channel is sufficient to keep the confidentiality of the secret. The process of initialization and sharing phase of the scheme are as follow:

1. The dealer chooses two large integer numbers *p* and *q*. Computes $N = pq$ such that $p \equiv 3 \bmod 4$ and $q \equiv 3 \bmod 4$.

2. Dealer selects an integer $g \in \left[N^{\frac{1}{2}}, N \right]$ such that *g* is relatively prime to *p*, *q* and publishes $\{g, N\}$.

3. Each participant chooses his/her secret shadow as $s_i \in [2, N]$ and computes $R_i = g^{s_i} \bmod N$.

4. Each participant sends his/her R_i to the dealer. Dealer ensures each R_i is unique, otherwise demands a new secret shadow from that participant.

5. Then, the dealer randomly chooses $s_0 \in [2, N]$ such that it is relatively prime to $(p-1)$ and $(q-1)$. Dealer also computes $R_0 = g^{s_0} \bmod N$ and publishes R_0.

6. Dealer computes $x_i = R_i^{s_0} \bmod N$ for each participant.

The dealer assigns a x_i - value for each participant. The participant can calculate his own x_i independently from secret shadow from publicly known value R_0 as

$$x_i = R_0^{s_i} \bmod N$$

Zhao, *et al.* (2007) also used the similar algorithm for secure distribution in their proposed multi-secret sharing scheme [8].

In our scheme, RSA (by Rivest, *et al.* 1978) algorithm [20] is used such that dealer encrypt each *x*-value by the public key of the participant and the *x*-value can be decoded by the corresponding participant (private key holder) only. The plain-text message (here the *x*-value) need to be preprocessed so that security can be ensured by random oracle model [13].

Random oracle model [13] was first established in 1993 by M. Bellare and P. Rogaway. In this paradigm, the authors stated that a practical protocol is produced by first devising and providing a protocol for the random oracle model; and then replacing the random oracle by appropriate hash function. The authors argued that this paradigm yields protocols those are much more efficient than standard ones and the paradigm applicable for encryption, signatures and zero-knowledge proof.

Bellare, & Rogaway (1995) proposed Optimal Asymmetric Encryption Protocol (OAEP) [14] for RSA, which is proven in the random oracle model. The OAEP algorithm uses a pair of random oracle *G* and *H* to preprocess the pain text before asymmetric encryption applied. The pair of oracle combined with trapdoor one way permutation function is semantically secure under plain-text attack (IND-CPA). If the scheme further can be combined with trapdoor function like RSA is also proven secure against chosen cipher attack (NM-CPA). The instantiation of G-Oracle for IND-CCA attack and instantiation of H-Oracle and use trapdoor permutation (trapdoor function) for NM-CPA attack are discussed by

Boldyreva and Fischlin (2006) in [1]. The authors also provided a detail security analysis of OAEP with random oracle model.

OAEP Algorithm for Preprocessing the Pain-text

Consider the following:

- f is a k-bits to k-bits trapdoor function (in our scheme RSA).
- k_0 is chosen such that adversary running time is significantly smaller than s^{k_0}.
- Length of the message x is $k-k_0$ (if the message is smaller then has to be padded with zeros).
- G is a "generator" such that $G:\{0,1\}^{k_0} \to \{0,1\}^n$ and H is the hash function $H:\{0,1\}^n \to \{0,1\}^{k_0}$.

Encoding:

1. Chose a random $r=\{0,1\}^{k_0}$.
2. Generate the encoded stream $\mathrm{E}=\left(x\oplus G\left(r\right)\right)\|\left(r\oplus H\left(x\oplus G\left(r\right)\right)\right)$, where $\|$ is concatenation operator. We can also represent the equation as $\mathrm{E}=X\|Y$ where and $Y=r\oplus H\left(x\oplus G\left(r\right)\right)$.

Decoding:

1. Extract X and Y parts from encoded message E and recover r as $r=Y\oplus H\left(X\right)$.
2. Recover the padded message $x=X\oplus G\left(r\right)$.

In proposed model, we use RSA as trapdoor function.

4. Proposed Scheme

We first present the secure and verifiable version of Shamir's (t,n)- threshold secret sharing as proposed in [2].

The objectives are defined as follows:

1. In the proposed version the *x*-values should not be predictable by the malicious users.
2. The shares can be distributed on insecure or public channels, but it can only be used by authorized users.
3. The shares are submitted by the participants must be verifiable, such that any fake share(s) submitted can be easily identified by the combiner.
4. The scheme should also identify if the dealer himself is dishonest and supplied fake shares to one or more participants.

The different phases of the model are as follows:

Initialization Phase

1. Each participant $P_i, i=1,2,...,n$ considers two large primes p_i, q_i and computes the following:

$$N_i = p_i q_i \text{ and } \varphi_i = (p_i-1)(q_i-1).$$

2. The participant $P_i, i=1,2,...,n$ chooses an integer $e_i, 1\le e_i \le \varphi$, such that $\gcd(e_i,\varphi_i)=1$.
3. Each participant also computes the secret exponent $d_i, 1\le d_i \le \varphi_i$, such that $e_i d_i \equiv 1 \bmod \varphi$.

The public keys are $Kpu_i=(e,N_i)$ and private keys are $Kpr_i=(d_i,p_i,q_i)$. Participant keeps $d_i, p_{i,}, q_i$ and φ_i secret.

4. Participants share the public keys - $Kpu_1, Kpu_2, ..., Kpu_n$ with the dealer and combiner.

Let, H is a suitable collision resistance one-way hash function, which takes an input as a binary string of variable length and outputs a hash-code which is a binary string with fixed length.

Construction of Shares

1. Dealer (*D*) uses a suitable random number generation function to generate n distinct random integers - $r_1, r_2, ..., r_n$.
2. *D* considers a polynomial function of Shamir's (t,n)-threshold secret sharing is in following form

$$F(x)=\left(S+a_1 x+a_2 x^2+....+a_{t-1}x^{t-1}\right) \bmod p$$

Then, the intermediate shares are computed as

$$s_1=(r_1,F(r_1)), s_2=\left(r_2,F\left(r_2\right)\right), ..., s_n=\left(r_n,F\left(r_n\right)\right)$$

3. *D* encrypts the random numbers r_i with the public keys Kpu_i as follows:

$$Er_i = r_i^{e_i} \bmod N \text{ for } i=1,2,...,n$$

4. *D* applies hash function *H* to generate hash code h_i for each $\left(Er_i,F\left(r_i\right)\right)$ as

$$h_i = H\left(Er_i,F\left(r_i\right)\right) \text{ for } i=1,2,...,n$$

5. *D* distributes the shares as follows:

$$sh_i=\left(Er_i,F\left(r_i\right)\right) \text{ for } i=1,2,...,n$$

6. *D* also applies the hash-function *H* on the secret *S* as - $h_s=H(S)$. Dealer publishes h_s.
7. *D* also publishes $h_1, h_2, ..., h_n$.

Recovery of Secret

1. The participant - P_i extracts Er_i out of his/her sh_i and decrypts it as

$$r_i = Er_i^{d_i} \bmod N.$$

2. P_i submits $sh_i=\left(r_i,F\left(r_i\right)\right)$ to the combiner.

3. If combiner (*C*) obtains t or more ($\le n$) shares in form of $\left(r_i, F\left(r_i\right)\right)$, he/she can applies Shamir's secret sharing to interpolate the value of secret $S = F\left(0\right)$.

Verifications

1. After receiving the share from *D*, each participant - P_i can verify if the obtained share has been modified by any intruder at the public channel. Participant P_i applies the hash-function to his/her share sh_i to obtain the hash-code h_i^* and confirms $h_i^* \stackrel{?}{=} h_i$. If $h_i^* \neq h_i$, then sh_i has been modified.

2. The combiner or participants can regenerate the hash-code for the obtained secret S^* as - $h_s^* = H\left(S^*\right)$ and verify with h_s as $h_s^* \stackrel{?}{=} h_s$ to confirm that it is the actual secret. If it does not match, then either one or more participants has supplied faked shares or the dealer is the dishonest (has supplied false share(s)). To verify the shares, combiner (*C*) can regenerate $sh_i = \left(Er_i, F\left(r_i\right)\right)$ from $\left(r_i, F\left(r_i\right)\right)$ using the corresponding public key (Kpu_i). *C* applies hash function *H* to obtain the hash-code h_i^*. If $h_i^* \neq h_i$ then the S_i share is faked and P_i is the identified cheater. If none of the shares is faked, then the dealer is identified as dishonest.

Proposed Model for Multi-Secret Sharing

Our $\left(t, k, n\right)$ – threshold multi-secret sharing scheme assumes the following:

- The k secrets are - $S_1, S_2, ..., S_k$.
- The n participants are - $P_1, P_2, ..., P_n$.
- *H* is a suitable, collision resistant, one-way hash function.

Initialization Phase

1. Each participant P_i, ($i = 1, 2, ..., n$) considers two large prime p_i, q_i and computes the following:

$$N_i = p_i q_i \text{ and } \varphi_i = \left(p_i - 1\right)\left(q_i - 1\right).$$

2. Each P_i chooses an integer $e_i, 1 \le e_i \le \varphi$, such that $\gcd\left(e_i, \varphi_i\right) = 1$.

3. Each P_i also computes the secret exponent $d_i, 1 \le d_i \le \varphi_i$, such that $e_i d_i \equiv 1 \bmod \varphi$.

The public keys are $Kpu_i = \left(e, N_i\right)$ and private keys are $Kpr_i = \left(d_i, p_i, q_i\right)$ for $i = 1, 2, ..., n$. P_i keeps $d_i, p_{i,}, q_i$ and φ_i secret.

4. Participants $P_i, i = 1, 2, ..., n$ share the public keys - $Kpu_1, Kpu_2, ..., Kpu_n$ with the dealer (*D*) and combiner (*C*). *D* ensures the public keys are unique otherwise demand a new public key.

Construction Phase

1. Dealer *D* chooses a suitable random number generator and generates random numbers $r_1, r_2, ..., r_n$.

2. *D* applies the hash-function *H* on the secrets as

$$h_{s_i} = H\left(S_i\right) \text{ for } i = 1, 2, ..., k$$

Concatenate all h_{s_i} to compute h_s as

$$h_s = \left(h_{s_1} \| h_{s_2} \| ... \| h_{s_k}\right);$$

3. Consider the polynomial as follow

$$if\left(k \le t\right)$$

3.1. Construct a polynomial $F\left(x\right)$ of degree $\left(t-1\right)$:

$$F\left(x\right) = S_1 + S_2 x + ... + S_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + ... + a_{t-k} x^{t-1} \bmod p;$$

where p is the large prime number and $0 < S_1, S_2, ..., S_k, a_1, a_2, ..., a_{t-k} < p$.

3.2. *D* computes $y_i = F\left(r_i\right) \bmod p$ for $i = 1, 2, ..., n.$

$$if\left(k > t\right)$$

3.3. Construct a polynomial $F\left(x\right)$ of degree $\left(k-1\right)$:

$$F\left(x\right) = S_1 + S_2 x + ... + S_k x^{k-1} \bmod p;,$$

where p is a large prime and $0 < S_1, S_2, ..., S_k < p$.

3.4. *D* computes $y_i = F\left(r_i\right) \bmod p$ for $i = 1, 2, ..., n$ and also computes $F\left(i\right) \bmod p$ for $i = 1, 2, ..., k - t$.

3.5. *D* publishes $F\left(i\right) \bmod p$ for $i = 1, 2, ..., k - t$.

4. Dealer encrypts the random numbers r_i with the public keys Kpu_i as follows:

$$Er_i = r_i^{e_i} \bmod N \text{ for } i = 1, 2, ..., n$$

5. Use hash function *H* to generate hash code h_i for each $\left(Er_i, y_i\right)$.

$$h_i = H\left(Er_i, y_i\right) \text{ for } i = 1, 2, ..., n$$

6. *D* publishes ($h_s, h_1, h_2, ..., h_n$).

7. *D* distributes the shares as follows:

$$sh_i = \left(Er_i, y_i\right) \text{ for } i = 1, 2, ..., n$$

Recovery Phase

1. Participant (P_i) extracts Er_i from sh_i and decrypts it as

$$r_i = Er_i^{d_i} \bmod N.$$

2. Submit $\left(r_i, y_i\right)$ to the combiner (*C*).

3. If combiner (*C*) obtains all t or more ($\le n$) shares in form of $\left(r_i, y_i\right)$, then the k secrets can be revealed by the reconstructing the unique polynomial $F\left(x\right)$ as follows:

$$if\ \left(k \le t\right)$$

$$F(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j\ne i}^{t} \frac{x - r_j}{r_i - r_j} \bmod p$$
$$= S_1 + S_2 x + ... + S_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + ... + a_{t-k} x^{t-1} \bmod p$$

$$if\ \left(k > t\right)$$

$$F\left(x\right) = \sum_{i=1}^{t} y_i \prod_{j=1, j\ne i}^{t} \frac{x - r_j}{r_i - r_j} \bmod p$$
$$+ \sum_{i=1}^{k-t} F\left(i\right) \prod_{j=1, j\ne i}^{k-t} \frac{x - j}{i - j} \bmod p$$
$$= S_1 + S_2 x + ... + S_k x^{k-1} \bmod p$$

Verifications

1. After receiving the share (sh_i^*) from the dealer, each participant (P_i) can verify $H\left(sh_i^*\right) \stackrel{?}{=} h_i$ to satisfy that the share is not modified or corrupted on the channel.

2. As t or more ($\le n$) shares (sh_i^*) are pooled by the combiner (*C*). *C* can reconstruct the k secrets as $S_1^*, S_2^*, ..., S_k^*$ and also calculates $h_s^* = H\left(S_1^*\right) || H\left(S_2^*\right) || ... || H\left(S_k^*\right)$. Thus, *C* can verify that the obtained secrets are the actual secrets or not by examining $h_s^* \stackrel{?}{=} h_s$.

If the obtained secrets are not true secrets ($h_s^* \ne h_s$), then either a few participants are cheaters or the dealer (*D*) is dishonest. The combiner can verify $H\left(sh_i^*\right) \stackrel{?}{=} h_i$ (as *C* possesses the public key Kpu_i for P_i, so *C* can recalculate sh_i^* from obtained $\left(r_i^*, F\left(r_i\right)_i^*\right)$) to ensure that the shares submitted by the participants are not faked. If $H\left(sh_i^*\right) \ne h_i$, then sh_i is faked and P_i is the cheater. But in the case $H\left(sh_i^*\right) = h_i$ for all $i = 1, 2, ..., t$, then the dealer (*D*) must be the dishonest and provided fake shares to one or more participants.

5. Performance

The performance features for the scheme proposed in [2] are as follow:

Robustness – A $\left(t, n\right)$ – threshold secret sharing scheme is called robust if any t or more ($\le n$) participants can reconstruct the full secret information. Shamir's secret sharing already holds this property.

Confidentiality - A $\left(t, n\right)$ – threshold secret sharing scheme holds confidentiality, if no information about the secret can be obtained if less than t shares are pooled.

Shamir's secret sharing already holds the property, along with that, the proposed scheme also promises that any malicious user learned the encrypted shares on the public channel had no clue about the *x*-values. Hence, malicious users or intruders cannot obtain a useful share.

Traceability - A $\left(t, n\right)$ – threshold secret sharing holds traceability, if it is able to detect any participant P_i who sends a fake share $S_i^* \ne S_i$ to the combiner. The proposed method utilizes hash function to make the shares verifiable.

Participant authentication and verification of shares – In the proposed scheme only a legal participant holds the legal private key to access his/her share (as the share is encrypted by corresponding public key by the dealer).

No need to have a secure channel for distribution – as the *x*-values are encrypted with the pubic keys before distribution, only the legal participants (the one possess the private key) can receive the *x*-values.

Verifiable secret – Hash-function ensures the correctness of the secret, obtained from the shares.

Comparisons between different secret sharing schemes based on verifiability of shares and secrets are shown in Table 1.

Our proposed verifiable $\left(t, k, n\right)$ – threshold multi-secret sharing scheme supports all the above mentioned attributes. As the MSS scheme is also based on Lagrange Interpolation, *robustness* and *confidentiality* are assured. If t or more ($\le n$) shares are obtained the all k secrets can be revealed. Otherwise (with the number of shares $< t$) none of the secrets can be achieved.

The traceability property is assured by use of hash-function.

The proposed scheme is also *t-consistent*, i.e. if the share s_i^* received by participant P_i passes the validity checking, it is guaranteed to be the valid ith share ($s_i^* = s_i$) for the k given secrets. The *t-consistent* property also assured by use of hash-function.

Table 1. Comparison between secret sharing schemes.

Scheme by	Secure distribution	Verification of shares	Verification of secret
Zhao, *et al.* in [21] (2009)	Yes	No	No
Harn, *et al.* in [11] (2009)	No	Yes	No
Ulutas, *et al.* in [18] (2011)	Yes	Yes	No
Harn, *et al.* in [12] (2014)	No	Yes	Yes
Liu, *et al.* in [22] (2015)	No	Yes	Yes
Liu, *et al.* in [23] (2016)	No	Yes	Yes
Our proposed scheme	Yes	Yes	Yes

6. Conclusions

Shamir's secret sharing scheme requires a secure channel to deliver the shares to the participants because it cannot resist attacks on the distribution channel. If the shares are communicated through insecure channel, then malicious users may learn the shares and with sufficient shares may reconstruct the secret. The same is true for any multi-secret sharing scheme which is based on Shamir's secret sharing scheme. In our proposed multi-secret sharing scheme, we have employed a method where some parts of the shares are communicated in encrypted form over a public channel. The shares only can be decrypted by authorized participants. Our proposed scheme also possesses properties to verify, whether the shares received by shareholders are consistent under the condition the secrecy of shares and secret both are maintained.

References

[1]. A. Boldyreva, M. Fischlin, On the Security of OAEP, in *Proceedings of the Advances in Cryptology – ASIACRYPT'06*, Vol. 4284, 2006, pp. 210-225.

[2]. A. K. Chattopadhyay, A. Nag, K. Majumder, A Verifiable and Cheating Resistant Secret Sharing Scheme, in *Proceedings of the International Conference on Advancement of Computer Communication & Electrical Technology (ACCET'16)*, Berhampore, India, October, 2016, pp. 264-268.

[3]. A. Shamir, How to Share a Secret, *Communications of the ACM*, Vol. 22, Issue 11, 1979, pp. 612-613.

[4]. C. Asmuth, J. Bloom, A Modular Approach to Key Safeguarding, *IEEE Transactions on Information Theory*, Vol. 29, Issue 2, 1983, pp. 208-210.

[5]. C.-C. Thien, J.-C. Lin, Secret image sharing, *Computers and Graphics*, Vol. 26, Issue. 5, 2002, pp. 765-770.

[6]. C.-C. Yang, T.-Y. Chang, M.-S. Hwang, A *(t, n)* multi-secret sharing scheme, *Applied Mathematics and Computation*, Vol. 151, Issue 2, 2004, pp. 483-490.

[7]. G. R. Blakley, Safeguarding cryptographic keys, in *Proceedings of the International Workshop on Managing Requirements Knowledge*, New York, June, 1979, pp. 313-317.

[8]. J. Zhao, J. Zhang, R. Zhao, A practical verifiable multi-secret sharing scheme, *Computer Standards & Interfaces*, Vol. 29, Issue 1, 2007, pp. 138-141.

[9]. J. Shao, Efficient verifiable multi-secret sharing scheme based on hash function, *Information Sciences*, Vol. 278, 2014, pp. 104-109.

[10]. J. Shao, Z. Cao, A new efficient *(t, n)* verifiable multi-secret sharing (VMSS) based on YCH scheme, *Applied Mathematics and Computation*, Vol. 168, Issue 1, 2005, pp. 135-140.

[11]. L. Harn, C. Lin, Detection and identification of cheaters in *(t, n)* secret sharing scheme, *Designs, Codes and Cryptography*, Vol. 52, No. 1, 2009, pp. 15-24.

[12]. L. Harn, M. Fuyou, C-C. Chang, Verifiable secret sharing based on the Chinese remainder theorem, *Security and Communication Networks*, Vol. 7, Issue 6, 2014, pp. 950-957.

[13]. M. Bellare, P. Rogaway, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, in *Proceedings of the 1st ACM Conference on Computer and Communications Security (CSS'93)*, New York, NY, USA, 1993, pp. 62-73.

[14]. M. Bellare, P. Rogaway, Optimal Asymmetric Encryption - How to Encrypt with RSA, in *Proceedings of the Eurocrypt'94*, 1995, pp. 1-19.

[15]. M. H. Dehkordi, S. Mashhadi, An efficient threshold verifiable multi-secret sharing, *Computer Standards & Interfaces*, Vol. 30, Issue 3, 2008, pp. 187-190.

[16]. M. H. Dehkordi, S. Mashhadi, New efficient and practical verifiable multi-secret sharing schemes, *Information Sciences*, Vol. 178, Issue 9, 2008, pp. 2262-2274.

[17]. M. Mignotte, How to share a secret, in *Proceedings of the Workshop on Cryptography*, Burg Feuerstein, Germany, March-April, 1982, pp. 371-375.

[18]. M. Ulutas, G. Ulutas, V. V. Nabiyev, Medical image security and EPR hiding using Shamir's secret sharing scheme, *Journal of Systems and Software*, Vol. 84, No. 3, 2011, pp. 341-353.

[19]. R.-J. Hwang, C.-C. Chang, An on-line secret sharing scheme for multi-secrets, *Computer Communications*, Vol. 21, Issue 13, 1998, pp. 1170-1176.

[20]. R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120-126.

[21]. R. Zhao, J.-J. Zhao, F. Dai, F.-Q, Zhao, A new image secret sharing scheme to identify cheaters, *Computer Standards & Interfaces*, Vol. 31, Issue 1, 2009, pp. 252-257.

[22]. Y. Liu, L. Harn, C.-C. Chang, A novel verifiable secret sharing mechanism using theory of numbers and a method for sharing secrets, *International Journal of Communication Systems*, Vol. 28, No. 7, 2015, pp. 1282-1292.

[23]. Y. Liu, C. Chang, An Integratable Verifiable Secret Sharing Mechanism, *International Journal of Network Security*, Vol. 18, No. 4, 2016, pp. 617-624.