A Secret Sharing Scheme for Preventing the Cheaters from Acquiring the Secret

Hassan Jameel, Sungyoung Lee

Department of Computer Engineering, Kyung Hee University
449-701 Suwon, Republic of Korea
{ hassan, sylee }@oslab.khu.ac.kr

Abstract. In this paper, we propose a secret sharing scheme which prevents the cheaters from recovering the secret when the honest participants cannot, with high probability. The scheme is a *(k, n)* threshold scheme providing protection against less than *k* cheaters. It is efficient in terms of share sizes for the participants. Furthermore the total size of the individual shares per participant is less than twice the size of the secret itself. The cheaters can do successful cheating with a probability *1/t*, which can be adjusted without significantly increasing the total size of the individual shares. Such a scheme can be deployed in thin client fat server systems where the server has reasonable computational power and there is a high level of mistrust among the users.

1. Introduction

Shamir[1] and Blakeley[2] independently proposed the notion of a secret sharing scheme. A (k, n) secret sharing scheme (threshold scheme) is a method of dividing a secret s into n shares $s_1, s_2, \ldots, s_n$ and giving a unique share to a set of n participants $\mathrm{P} = \{P_1, P_2, \ldots, P_n\}$ such that the knowledge of any k or more shares makes the secret s easily computable but knowledge of $k-1$ or less shares does not reveal any information about the secret s. Here, k is called the threshold of the scheme. The secret s is chosen by a Dealer $D \notin \mathrm{P}$ and gives the share s_i to the participant P_i over a secure channel.

An elegant feature of Shamir's Scheme is that it is unconditionally secure. A consequence of this property is that the size of shares per participant be at least the size of the original secret [3]. This poses a fundamental limit on unconditionally secure secret sharing schemes. Against resource bounded adversaries this lower limit can be relaxed by introducing computational security. Krawczyk [18] was the first to discuss computationally secure secret sharing schemes by reducing the size of shares to approximately $|s|/k$. However our discussion will be limited to unconditionally secure schemes. Although Shamir's secret sharing scheme seems perfectly secure, however it is not free of conspiracies and cheating. Tompa and Woll[5] presented a possible scenario in which a participant or a group of participants can send a false share and hence get the correct secret whereas the honest participants will be deceived with the wrong secret. This situation will go undetected in Shamir's scheme described above. Tompa and Woll[5] first showed how to prevent this form of cheating keeping in mind the unconditional security assumption. An outcome of their scheme is that the size of shares should grow with the probability of cheating detection. This was shown to be undeniable in [14] where the authors gave lower bound on the size of shares. This bound was improved by Kurosawa et al in [20]. Another undesirable property of cheating detection

schemes is that even though cheating is detected (or in some other schemes cheaters are identified) the cheaters still recover the secret whereas the honest participants don't get the correct secret. Tompa and Woll showed how to prevent this situation. They encoded the secret as a sequence, where only one element of the sequence is assigned the actual secret and the other elements a dummy value. These elements are then divided into shares and given to each participant. There are a couple of disadvantages in their method. Firstly, each participant has t shares to keep (equal to the length of the sequence). Secondly the probability of the cheaters to get the secret is $\approx 1/t$ which can be made less by making the number of shares large, where each share is at least the size of the original secret. The goal of this paper is to increase this probability without considerably increasing the size of shares.

Related Work: A number of cheater detection and identification techniques have been proposed [5, 6, 7, 8, 9]. Some others are given in [20,22,23,24]. A serious drawback of their results is the dramatic expansion of shares as discussed by Piepryzk and Zhang in [25]. They propose a cheating detection scheme for a general linear secret sharing scheme (Shamir's Scheme is a special case of such schemes). Amazingly their scheme does not involve any share expansion. However, it uses shares from the extension field $GF\left(p^{v}\right)$ instead of the usual prime field $GF\left(p\right)$. Secondly and more importantly, in order to detect cheaters, the actual number of participants involved is greater than the threshold of the scheme. As the name of these schemes suggests, they only detect or identify the set of cheaters and do not stop them from recovering the secret whereas the honest participants still don't get the correct secret. This property can be achieved by using verifiable secret sharing schemes. As an example see the scheme proposed by Rabin and Ben-Or [21]. Such schemes have capabilities of verifying the shares distributed by a possibly dishonest dealer and verifying the shares of the participants during secret reconstruction phase before actually revealing the secret. This verification generally involves digital signatures and pooling of encrypted shares and hence these schemes are only computationally secure. Notice that the cheaters won't get the secret too because their shares are rejected before combining the secret shares. However, the communication model of these schemes is different from the one addressed in this paper. First of all we do not assume a dishonest Dealer. Secondly, on secret reconstruction we assume that all participants are able to "see" the unencrypted pooled shares. Thirdly, we assume a simultaneous pooling of shares. Under these assumptions, a dishonest participant would still recover the secret as he would now have the knowledge of the pooled shares.

Lin and Harn [10] proposed a method that can be deployed under our assumptions for cheating prevention. Still, in their scheme, the probability of the cheater recovering the secret is inversely proportional to the number of shares. Lee and Laih[11] proposed the notion of a $\left(v,k,n\right)$ fairness secret sharing scheme in which there are $v < k/2$ cheaters and the probability of recovering the secret is equal for all participants. Their scheme was improved in efficiency by Hwang and Chang [12]. They use two polynomials to hide two seemingly unrelated numbers necessary to find the secret. Knowledge of one of the numbers leaves absolutely no information about the other and hence the secret. Their

scheme requires each participant to keep 2 shares each the size of the actual secret and is unconditionally secure. However they assume that the number of cheaters is less than $k/2$, which in most cases may not be true. Indeed in Tompa and Woll model the number of cheaters can be up to $k-1$. More recently Pieprzyk and Zhang [19] have discussed such cheating immune schemes where the secret is from $GF\left(p^{v}\right)$. We will focus on a cheating immune system where the secret and all the shares are from the prime field $GF\left(p\right)$. In this paper we propose a secret sharing scheme to prevent cheating with high probability using individual shares whose total size is roughly the same as the size of the secret. The basic theme is to divide the shares into parts, introduce random numbers and shuffle them with the shared pieces in such a way that the participants don't know the exact construction of the shares unless at least k of them pool their shares together. The participants pool the individual parts one by one and the notion of hash functions as in [4] is used to detect cheating at every stage. The resulting scheme prevents cheating against any set of $k-1$ cheaters. Introduction of a one way hash function of course takes away the unconditionally secure property of the Shamir Secret Sharing Scheme, however it can help in reducing the size of the shares per participants. The scheme can successfully detect cheating provided there exists a collision free one way hash function. The main contribution of our work is not to detect the cheaters but to prevent them from attaining the secret under our communication model. The rest of the paper is organized as follows: In Section 2, Shamir's Secret Sharing scheme is introduced. Section 3 discusses cheating in Shamir's scheme. Section 4 describes the use of one way hash functions for cheating detection. In Section 5 we propose our scheme. Section 6 gives the security analysis followed by computational analysis in Section 7 and finally we conclude in Section 8.

2. Shamir's Secret Sharing

In this section, we describe Shamir's[1] Secret Sharing scheme more elaborately. Given a set P of n participants $\left\{P_1, P_2, \ldots, P_n\right\}$, a set $S = \{0,1,2,\ldots,s-1\}$ of possible values of the secret and the threshold k of the scheme:

1. Choose a prime $p \geq \max\left\{s, n\right\}$.
2. Let $\mathbb{Z}_p^*$ be the field of non negative integers modulo p.
3. Randomly choose $a_{1,}a_2,\ldots,a_{k-1}$ from a uniform distribution over the integers $\left[0, p\right)$.
4. Choose the secret a_0 from the set S.
5. Construct the polynomial $f\left(x\right) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1}x^{k-1}$.
6. $\forall\ 1 \leq i \leq n$, let $s_i = f(i) \bmod p$.
7. Give the share s_i to the participant P_i over a secure channel.

Any k out of n participants can pool their shares together to construct the polynomial $f\left(x\right)$ by langrange's polynomial interpolation to get a_0, which when $x = 0$ is given by:

$$f(0) = a_0 = \sum_{i=1}^{k} s_i \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{j}{j-i} \bmod p$$

However if the participants are less than k, they cannot reconstruct the polynomial $f(x)$ and hence do not get any information about $f(0) = a_0$. The scheme involves two phases: Secret Generation and Secret Recovery.

3. Cheaters in Shamir's Scheme

Although the scheme described in section 2 seems "perfectly" secure, however it is not secure against cheating. Tompa and Woll[5] pointed out a possible cheating scenario. Without loss of generality suppose participants $P_1, P_2, \ldots, P_k$ decide to pool their shares together and P_1 decides to cheat. P_1 interpolates a $k-1$ degree polynomial $\Delta(x)$ such that $\Delta(0) = -1$ and $\Delta(i) = 0$ for $2 \le i \le k$. He then gives the share $s_1 + \Delta(1)$ instead of the given share $s_1 = f(1)$. The k participants will now construct the polynomial $f(x) + \Delta(x)$ with the constant term $f(0) + \Delta(0) = a_0 - 1$. All the honest participants will get the wrong secret where as the cheater can recover the correct secret by subtracting $\Delta(0)$ from the final result. Furthermore this cheating will go undetected.

Tompa and Woll[5] proposed a cheating detection model in which each of the n participant is given the share $S_i = (x_i, s_i)$, where $(x_1, x_2, \ldots, x_n)$ are chosen uniformly and randomly from all permutations of n distinct elements from $\{1, 2, \ldots, p-1\}$, and $s_i = f(x_i)$. As the cheaters do not know the x_i's of the other participant they cannot cheat in the way mentioned above without being undetected with the probability about $(s-1)(k-1)/(p-k)$. However, although cheating is detected with high probability, the cheaters cannot be identified. Furthermore, the cheaters can still construct the secret and the honest one's cannot. To solve the latter problem, they encode the secret s into a sequence $s^{(1)}, s^{(2)}, \ldots, s^{(t)}$ for some t. They set $s^{(i)} = s$ for some i chosen randomly and $s^{(j)} = w$ for all $j \neq i$ where w is a dummy integer. The $s^{(i)}$'s are divided into shares and given to each participant. During secret recovery phase, the participants pool all their shares together and interpolate the $s^{(i)}$'s one at a time until they get some $s^{(i)} \neq s$. If this $s^{(i)}$ is not legal, then they know that cheating has occurred. The probability of cheating on this moment is $\approx 1/t$. To make this probability small, the value of t should be large enough which implies increase in the number of shares per participant. Moreover the size of each share should be equal to the size of the actual secret. We would like to find out a cheating prevention scheme in collaboration with a cheating detection and identification scheme in which the share sizes are not significantly affected by the probability of successful cheating.

Before we present our scheme we would like to illustrate the assumptions used in our scheme:

A1. There is an honest Dealer
A2. When the shares are pooled during Secret Recovery, they can be viewed by all the participants involved in the process. In other words they no longer remain "secret".
A3. During secret recovery, all participants will pool their share simultaneously. (This avoids the situation in which a participant deliberately holds its share if he is the last one to pool and hence can see all the other shares which can be used in secret recovery).

4. Using One Way Hash Functions to Detect Cheating

For cheating detection and identification, T.C. Wu and T. S. Wu [4] used the idea of one way hash function and arithmetic coding. Any one way hash function $h(.)$ can be used with the usual properties of collision freeness, input of any length and output of a fixed length. Their proposed method is based on the following two theorems whose proof can be found in [13].

Theorem 4.1 If $T = \sum_{i=1}^{n} a_i p^{i-1}$, where $0 \le a_i < p$, then

$$\left\lfloor \frac{T}{p^{i-1}} \right\rfloor \bmod p = a_i, \text{ for } 1 \le i \le n .$$

From Theorem 4.1, we get the following general result:

Theorem 4.2 If $T = \sum_{i=1}^{n} a_i p^{2(i-1)} + \sum_{i=1}^{n-1} cp^{2i-1}$, where $-p < a_i < p$ and $1 \le c < p$, then

$$\left\lfloor \frac{T}{p^{2(i-1)}} \right\rfloor \bmod p = a_i \bmod p, \text{ for } 1 \le i \le n$$

For the share s_i of participant P_i, $h(s_i)$ is calculated, where $h(.) < p$. For a positive constant $c < p$, the following is computed.

$$T = \sum_{i=1}^{n} h(s_i)\, p^{2(i-1)} + \sum_{i=1}^{n-1} cp^{2i-1}$$

The values T and p are made public. Now upon reconstruction of the secret, the share s_i' of participant P_i is checked as follows. P_i calculates $T' = \sum_{i=1}^{n} h(s_i')\, p^{2(i-1)}$. It is checked whether $\left\lfloor \frac{T - T'}{p^{i-1}} \right\rfloor \bmod p = 0$. If this is not the case then P_i is a cheater.

The check for cheating detection and identification follows from Theorem 4.2 and the collision free property of $h(.)$.

As the name identifies, this is a cheating detection and identification technique. So, under our assumptions the cheaters can still get the secret even if they are identified (as all participants have pooled their shares). It would be nice to modify this technique for cheating prevention too. Of interest would be the outcome that the share sizes of the participants be almost as much as the size of the secret itself.

5. Proposed Scheme

Before going on to the proposed scheme, we revise the notion of permutations which will be needed to describe our scheme. For a positive integer t, let $[t] = \{1, 2, \ldots, t\}$. A *permutation* σ of $[t]$ is a one-to-one correspondence from $[t]$ to $[t]$. Let R_t denote the set of all permutations of $[t]$. The identity of R_t denoted by ι is a permutation defined by the rule $\iota(x) = x, \forall x \in [t]$. The composition of two permutations σ and τ is defined as $\sigma\tau(x) = \sigma(\tau(x)), \forall x \in [t]$. The inverse $\sigma^{-1} \in R_t$ of the permutation $\sigma \in R_t$ is defined by $\sigma\sigma^{-1} = \sigma^{-1}\sigma = \iota$. A *ranking function* assigns a unique integer in the range $[1, t!]$ to each of the $t!$ permutations in R_t. The corresponding *unranking function* is the inverse, which given an integer in the range $[1, t!]$ returns the permutation with this rank.

Next we define a few functions which will be required in our scheme. Let p denote a positive integer and let $x \in \mathbb{Z}_p^+$. Let $|p|$ denote the number of bits in the binary representation of p.

Definition 5.1 The concatenation of the positive integers x and y is a number obtained by appending the binary representation of y after the binary representation of x. We denote it by $con(x, y)$.

This definition can be extended to more than two arguments in a natural way.

Definition 5.2 For a positive integer t, the function $split(x, t, p)$ is an ordered t-tuple defined by,

$$split(x, t, p) = (x_1, x_2, \ldots, x_t)$$

Such that,

$$|x_i| = \lceil |p|/t \rceil \text{ for } 1 \le i \le t-1,$$
$$|x_t| = |p| - \lceil |p|/t \rceil (t-1),$$
$$\text{and } con(x_1, x_2, \ldots, x_t) = x.$$

Definition 5.3 Given an ordered t-tuple of elements $x=(x_1,x_2,\ldots,x_t)$, we say that the ordered t-tuple $O_x(\sigma)=\left(x_{\sigma(i_1)},x_{\sigma(i_2)},\ldots,x_{\sigma(i_t)}\right)$ is a reordering of x according to the permutation σ, if

$$1=\sigma(i_1)<\sigma(i_2)<\ldots<\sigma(i_t)=t$$

where $i_1,i_2,\ldots,i_t\in[t]$ distinct from each other and *not necessarily* in ascending order.

We denote the initial ordered t-tuple $x=(x_1,x_2,\ldots,x_t)$ by $O_x(\iota)$.

Now we are set to describe our scheme. Continuing with our previous notation, let P be the set of n participants $\{P_1,P_2,\ldots,P_n\}$, let $S=\{0,1,2,\ldots,s-1\}$ be the set of possible values of the secret and let k be the threshold of the scheme. The scheme will be shown in two steps: Secret Generation Phase and Secret Recovery Phase. We assume an honest Dealer D and a secure channel between the Dealer and each participant in the Secret Generation Phase.

5.1 Secret Generation Phase

This phase is carried out by the Dealer D. It is assumed that for a given integer t, the set of all permutations R_t is ranked according to a ranking function.

1. Choose a prime $p\geq\max(s,n,t!)$.
2. Select the secret S from the set of non negative integers $\{0,1,2,\ldots,s-1\}$.
3. Select random integers $a_1,a_2,\ldots,a_n$ from the integers $(0,p)$ to construct the $k-1$ degree polynomial:
$$f_1(x)=S+a_1x+a_2x^2+\ldots+a_{k-1}x^{k-1}$$
4. Compute $s_i=f(i)$ for $1\leq i\leq n$.
5. Select a positive integer t, and compute $b=\lceil|p|/t\rceil$.
6. For $1\leq i\leq n$,
   6.1 Compute $split(s_i,t,p)$. If $\lceil|p|/t\rceil\neq|p|/t$, then select a random positive integer q such that $|q|=\lceil|p|/t\rceil t-|p|$ and compute $s_{it}=con(s_{it},q)$. Now concatenate the first $t-1$- pieces as $S_i=con\left(s_{i1},s_{i2},\ldots,s_{i(t-1)}\right)$.
   6.2Let $e_{i1}=s_{it}$ and select $t-1$ random positive integers $e_{i2},e_{i3},\ldots,e_{it}$ such that each one is of size $\lceil|p|/t\rceil$. Create the t-tuple $O_{s_i}(\iota)=(e_{i1},e_{i2},\ldots,e_{it})$.
7. Choose a one way function $h(.)$ with the condition that $h(.)<p$ and a positive constant c such that $1\leq c<p$.

8. For $1 \le j \le t$, compute $T_j = \sum_{i=1}^{n} h\left(e_{ij}\right) p^{2(i-1)} + \sum_{i=1}^{n-1} cp^{2i-1}$. Let $O_T\left(\iota\right) = \left(T_1, T_2, \ldots, T_t\right)$.
9. Compute $T = \sum_{i=1}^{n} h\left(S_i\right) p^{2(i-1)} + \sum_{i=1}^{n-1} cp^{2i-1}$.
10. Randomly select a permutation σ_x from R_t, where $1 \le x \le t!$ and compute its inverse permutation $\sigma_y = \sigma_x^{-1}$.
11. Use Shamir's secret sharing to evaluate a polynomial modulo a prime $p' \ge \max\left(n, t!\right)$ and $\left|p'\right| = b$ with shadows $d_1, d_2, \ldots, d_n$ and constant term y.
12. For $1 \le i \le n$, Compute:
    12.1. $c_i = e_{i1} \oplus e_{i2} \oplus \ldots \oplus e_{it} \oplus d_i$ and store in a public directory.
13. For $1 \le i \le n$, compute $O_{s_i}\left(\sigma_x\right)$ and send to the participant P_i along with S_i over a secure channel.
14. Compute $O_T\left(\sigma_x\right)$ and store it along with p and T in the public directory.

Notice that the inverse permutation σ_y is hidden in the second polynomial and hence the exact position of s_{it} in the t-tuple is not known to any participant as they don't know σ_x either, unless they pool all their shares together to evaluate the second polynomial.

5.2 Secret Recovery Phase

Without loss of generality assume that participants $P_1, P_2, \ldots, P_k$ decide to pool their shares.

1. Participants submit their first share S_i.
    1.1. For participant P_i, compute $T' = \sum_{i=1}^{n} h\left(S_i\right) p^{2(i-1)}$.
    1.2. For each P_i, check whether
    $$\left\lfloor \frac{T - T'}{p^{i-1}} \right\rfloor \bmod p = 0$$
    If the above equation does not hold, then P_i is a cheater.
    1.3. If at least one cheater has been detected and identified then terminate the phase here.
2. For $1 \le j \le t$:
    2.1. Participants submit their j th element of $O_{s_i}\left(\sigma_x\right)$.
    2.2. For participant P_i, compute $T' = \sum_{i=1}^{n} h\left(s'_{ij}\right) p^{2(i-1)}$.
    2.3. For each P_i, check whether

$$\left\lfloor \frac{T_j - T'}{p^{i-1}} \right\rfloor \bmod p = 0$$

If the above equation does not hold, then P_i is a cheater.

2.4. If at least one cheater has been detected and identified then terminate the phase here.

3. For $1 \le i \le k$:
   3.1. Compute $d_i = e_{i1} \oplus e_{i2} \oplus \ldots \oplus e_{it} \oplus c_i$.
4. Use Langrange's polynomial interpolation to find x from these d_i's and retrieve the permutation σ_y from R_t through the unranking function.
5. For $1 \le i \le k$:
   5.1. Apply the inverse permutation $\sigma_y = \sigma_x^{-1}$ to obtain $O_{s_i}\left(\sigma_x^{-1}\sigma_x\right) = O_{s_i}(\iota)$.
   5.2. If $\lceil |p|/t \rceil \ne |p|/t$, remove $\lceil |p|/t \rceil t - |p|$ last bits from the first element of $O_{s_i}(\iota)$.
   5.3. Compute $con(S_i, e_{i1}) = s_i = f(i)$.
6. Reconstruct the polynomial $f(x)$ from these k shares using Langrange's polynomial interpolation.
7. Recover the secret as $S = f(0)$.

At any stage of the recovery phase, if the cheaters change their elements in the t-tuples, the procedure will be terminated. And no one will submit the remaining elements of the t-tuples. It will be shown in the next section that the probability of successful cheating in this scheme depends upon the parameter t which can be adjusted to decrease this probability.

6. Security Analysis

The above scheme acts both as a cheating detection and prevention scheme. The check to detect and identify the cheaters in step 1.3 and 2.3 of the recovery phase is justified by the following theorem:

Theorem 6.1 If $\left\lfloor \frac{T_j - T'}{p^{i-1}} \right\rfloor \bmod p \ne 0$ in step 1.3 and 2.3 of the Secret Recovery Phase, then player P_i is a cheater.

Proof The proof is given in [4], however we describe it here for completeness. Now from Theorem 4.2,

$$\left\lfloor \frac{T_j - T'}{p^{i-1}} \right\rfloor \bmod p = h(s_j) - h(s'_j) \pmod p$$

As $h(.) < p$, this means that $h(s_j) - h(s'_j) \pmod p = 0$ if and only if $h(s_j) = h(s'_j)$. Since $h(.)$ is collision free, therefore $h(s_j) = h(s'_j) \Leftrightarrow s_j = s'_j$. Thus, if this value is not zero then player P_i submitted a false share and is a cheater.

Next theorem shows the bound for the share sizes for each participant in information theoretic terms. For a background in information theory we refer the reader to [15] for details. Let $H(P_i)$ represent the total size of shares held by each participant and let $H(S)$ represent the size of the secret and let $\ln(z) = |z|$ represent the size of the any positive integer where the base of $\ln(.)$ is 2.

Theorem 6.2 In the above scheme

$$H(S) \le \ln(p) + b(t-1) \le H(P_i) \le \ln(p) + \ln(t-1) + b(t-1)$$

Where $b = \lceil |p|/t \rceil$ and in the last expression of the inequality $t > 1$.

Proof Since, $p \ge \max(s, n, t!)$, so $\ln(S) \le \ln(p)$. Now, the last $t-1$ elements of the ordered t - tuples $O_{s_i}(\iota)$ with each participant have size: $b(t-1)$. If t divides $|p|$ i.e. $|p| \bmod t = 0$, then $\ln(S_i) + \ln(s_{ti}) = t\lceil |p|/t \rceil = t(|p|/t) = |p| = \ln p$. So in this case $\ln(P_i) = \ln p + b(t-1)$. Next, if $|p| \bmod t \ne 0$, then the size of $\ln(S_i) + \ln(s_{ti})$ is $\ln(p) + \ln(q)$, where $\ln q = |q| = \lceil |p|/t \rceil t - |p|$ as defined in the scheme. Let $|p| \bmod t = r$, then

$$\ln(q) = \lceil |p|/t \rceil t - |p| = \left\lceil \frac{|p| + t - r}{t} \right\rceil t - |p| = \left(\frac{|p| + t - r}{t} \right) t - |p| = t - r .$$

Which has the maximum value when $r = 1$. Thus, in this case: $\ln(S_i) + \ln(s_{ti}) = \ln(p) + \ln(t-1)$. In all other cases when $|p| \bmod t \ne 0$ it is clear that the inequality $\ln(S_i) + \ln(s_i) < \ln(p) + \ln(t-1)$ holds. So, if $|p| \bmod t \ne 0$, then $\ln(P_i) \le \ln p + \ln(t-1) + b(t-1)$. Now since the secret S and the random elements in the t-tuple are uniformly chosen, therefore $\ln|S| = H(S)$ and $\ln|P_i| = H(P_i)$. This proves the theorem.

Corollary 6.1 When $p = s$ and t divides $|p|$, the above scheme has a minimum total share size per participant given by $H(S) + \left(\frac{t-1}{t} \right) H(S)$.

Proof If $p = s$, then $H(S) = \ln(S) = \ln(p)$ and as seen above if t divides $|p|$ then $b = (|p|/t)$. So $\ln(P_i) = \ln(p) + b(t-1) = \ln p + (|p|/t)(t-1) = \ln p + (\ln p)\left(\frac{t-1}{t}\right)$. Thus,

$\ln(P_i) = H(S) + \left(\frac{t-1}{t}\right)H(S)$, which is obviously the minimum total share size attainable in our scheme.

We describe a modified Tompa and Woll attack, which will be shown to be more effective in our scheme in the next theorem. Without loss of generality suppose participants $P_1, P_2, \ldots, P_k$ decide to pool their shares together and P_1 decides to cheat. P_1 interpolates a k degree polynomial $\Delta(x)$ such that $\Delta(0) = -1$, $\Delta(1) = r$ for some chosen r and $\Delta(i) = 0$ for $2 \le i \le k$. He then computes $\Delta(k+1)$. From Langrange's polynomial interpolation we have:

$$\Delta(x) = \sum_{i=1}^{k+1} \Delta(i) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{x-j}{i-j}$$

$$= \Delta(1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{x-j}{i-j} + \Delta(k+1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{x-j}{i-j}$$

When $x = 0$, the above equation becomes:

$$\Delta(0) = \Delta(1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{j}{j-i} + \Delta(k+1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{j}{j-i}$$

$$= \Delta(1) \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{j}{j-i} \right) \frac{k+1}{k} + \Delta(k+1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{j}{j-i}$$

$$= \Delta(1) \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{j}{j-i} \right) + \Delta(1) \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{j}{j-i} \right) \left(\frac{1}{k} \right) + \Delta(k+1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{j}{j-i}$$

He then gives the share $s_1 + \Delta(1)$ instead of the given share $s_1 = f(1)$. The k participants will now construct a polynomial with the constant term $f(0) + \Delta(0) - \Delta(1) \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{j}{j-i} \right) \left(\frac{1}{k} \right) - \Delta(k+1) \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{j}{j-i}$. The cheater however can get the correct secret by subtracting the last three terms from the above result. The honest participants will get the wrong secret unless the above term is equal to zero.

Now we would like to find a bound for the probability of cheating in our scheme. The following theorem will also illustrate the benefit of the modified Tompa and Woll attack as described above.

Theorem 6.3 The proposed scheme prevents the modified Tompa and Woll attack with a probability $(t-1)/t$.

Proof Sketch First notice that if the cheater wishes to cheat during the pooling of S_i's (the first shares) then he will be detected in step 1.2 and the recovery phase will stop leaving the s_i's (the remaining shares) of the honest participants undetermined. This leaves him with no choice but to send his first share unmodified. Now only one of the elements of the t-tuple is the remaining piece of the original share s_i. As the cheater has already submitted his correct first share, so he has to submit his modified share such that it only differs from his original share s_i in the last $\lceil |p|/t \rceil$ bits. Suppose the cheater decides to cheat during the submission of the j th element of the t-tuple. Assuming that this is in fact the remaining piece of the share s_i i.e. $e_{ij} = s_{it}$, he does as follows: Using the modified Tompa and Woll attack as described above, compute $con(0, s''_{it})$, where $|0| = |S_i|$ and all its bits are zero. Now let

$$s_i + r = con(S_i, s''_{it}) \Rightarrow r = con(S_i, s''_{it}) - con(S_i, e_{ij}) = con(0, s''_{it} - e_{ij})$$

If $|p| \bmod t \neq 0$ then remove the last $\lceil |p|/t \rceil t - |p|$ bits from $s''_{it} - e_{ij}$. Use this r in the modified Tompa and Woll attack. Now, three cases might occur:

Case 1 The s_{it}'s have not yet been pooled. In this case no one can construct the secret as the last pieces of the individual shares have not been pooled.
Case 2 The cheater is lucky and $e_{ij} = s_{it}$. The probability that this might happen is $\frac{(t-1)!}{t!} = \frac{1}{t}$.
Case 3 The cheating is done after the s_{it}'s have been pooled. In this case, the cheater can also cheat to modify the hidden permutation during the submission of the last element of the t-tuple. However, whatever the situation may be, as the correct share has already been pooled, the honest participants can also construct the secret by trying at most $t-1$ different combinations of the correct pooled elements of the t-tuple and can eventually know the secret value.

The above proof can be generalized to any set of $k-1$cheaters. Notice that if any cheater or set of cheaters are detected without the submission of the s_{it}'s, the Dealer can redistribute new shares to the honest participants over their secure channels and discard the old secret, so that the cheaters cannot guess the honest participant(s) share by guessing the undetermined part. The situation in case 3 of the above theorem can be made

more convenient for the honest participants, if a validity check for the constructed secret is introduced. Such a check can also work as a cheating detection technique.

7. Computational Complexity

In the secret generation phase, any noticeable extra computational cost other than the threshold scheme is the computation of the one way hash function. The hash function has to be applied *nt* times on inputs of length $\lceil |p|/t \rceil$ and *n* times on inputs of length $(t-1)\lceil |p|/t \rceil$. As indicated in [4] for the hash function $h(.)$ to be collision free, the output should be at least 128 bits. Thus this phase also includes computation of T_j and T, $t+1$ times modulo 128 bits if we assume the prime p to be 128 bits. In addition to that, another polynomial evaluation modulo p' is required. However this computation can be done fairly quickly if we assume the Dealer to have enough computational resources.

In the secret construction phase, the hash function is employed *kt* times on inputs of length $\lceil |p|/t \rceil$ and *k* times on length $(t-1)\lceil |p|/t \rceil$ bits, and the check in step 1.3 requires $tk+k$ 128-bit modulo divisions and reductions. The extra computation cost caused by the hash function can be reduced if another deterministic cheating detection and identification technique can be employed whose efficiency does not depend upon the size of the input.

Finally, a note on the ranking of the permutations in R_t. Algorithms exist to compute both the ranking and unranking functions in $O(t)$ arithmetic operations or better. See, for example [16]. The reordering of the t - tuples can be done in $O(t \log t)$ worst case time[17] and the computation of the inverse permutation σ_x^{-1} can be done in $O(t)$ time.

8. Conclusion & Discussion

In this paper, we have proposed a cheating prevention scheme with an adjustable probability of cheating. The scheme employs dividing the secret share into parts and reshuffling these parts randomly such that no participants know the exact sequence. The resulting scheme has a probability of cheating proportional to the number of divided parts. The total size of individual shares is always less than twice the size of the secret. In a practical application, if the original secret is 512 bits, then the shares of participants can be divided into 17 parts, one 480 bits long and each of the remaining 16 parts 32 bits long. The probability of successful cheating is $1/16 \approx 0.0625$. It should be noted however that once the permutation σ has been revealed during the secret construction, cheating can be done straightforwardly in the next construction of the secret. Such a situation can be counteracted by using a black box implementation which is suitable for all secret sharing schemes. Computation wise this scheme is suitable for the thin client fat server scenario in which the server can do the more expensive computations.

9. References

[1] Shamir, A.: "How to Share a Secret," *Communication of ACM*, Vol. 22, 1979, pp. 612-613.

[2] Blakley, G. R. "Safeguarding Cryptographic Keys," *Proceedings of the National Computer Conference*, American Federation of Information Processing Societies, Vol. 48, 1979, pp. 242- 268.

[3] E.D. Karnin, J.W. Greene, M.E. Hellman, On secret sharing systems, IEEE Trans. Inform. Theory 29 (1983) 35-41.

[4] T.-C. Wu and T.-S. Wu, Cheating detection and cheater identification in secret sharing schemes, *IEE Transactions on Computers and Digital Techniques* **142** (1995), 367-369.

[5] M. Tompa , H. Woll, How to share a secret with cheaters, Journal of Cryptology, v.1 n.2, p.133-138, Aug. 1988.

[6] Carpentieri, M. "A Perfect Threshold Secret Sharing Scheme to Identify Cheaters," *Design, Codes and Cryptography*, Vol. 5, 1995, pp. 183-187.

[7] Cai, Ning and Lam, Kwok Yan, "On identification secret sharing schemes", *Information and Computation*, Vol. 184, 2003, pp. 298-310.

[8] Chang, C. C. and Hwang, R. J. "An Efficient Cheater Identification Method for Threshold Schemes," *IEE Proceedings-Computers and Digital Techniques*, Vol. 144, No. 1, 1997, pp.23-27.

[9] Wu, T. C. and Wu, T. S. "Cheating Detection and Cheater Identification in Secret Sharing Schemes," *IEE Proceedings-Computers and Digital Techniques*, Vol. 142, No. 5, 1995, pp.367-369.

[10] Lin, H. Y. and Harn, L.: "Fair Reconstruction of a secret," *Information Processing Letters*, Vol. 55, 1995, pp.45-47.

[11] Lee, Y. C. and Laih, C. S.: "A V-Fairness (*t*, *n*) Secret Sharing Scheme," *IEE Proceedings- Computers and Digital Techniques*, Vol. 144, No. 4, 1997, pp. 245-248.

[12] Ren-Junn Hwang, Chin-Chen Chang: Enhancing the Efficiency of (v, r, n)-Fairness Secret Sharing Scheme. AINA (1) 2004: 208-212.

[13] KNUTH, D.E., The Art of Computer Programming. Vol. 2, Semi numerical Algorithms. Addison-Wesley, 1981.

[14] M. Carpentieri, A. De Santis, U. Vaccaro, Size of Shares and Probability of Cheating in Threshold Schemes, Proceeding of Eurocrypt'93.

[15] T.M. Cover and J.A.Thomas, Elements of Information Theory. New York: Wiley 1991.

[16] Wendy J. Myrvold, Frank Ruskey: Ranking and unranking permutations in linear time. Information Processing Letters. Vol 79/6, pp 281-284, Sep 2001.

[17] Faith E. Fich , J. Ian Munro , Patricio V. Poblete, Permuting In Place, SIAM Journal on Computing, v.24 n.2, p.266-278, April 1995.

[18] Hugo Krawczyk, Secret sharing made short, Proceedings of the 13th annual international cryptology conference on Advances in cryptology, p.136-146, January 1994, Santa Barbara, California, United States

[19] J. Pieprzyk, X.-M. Zhang, Cheating Prevention in Secret Sharing over GF(p^t), Indocrypt 2001, LNCS 2247, pp. 79-90, 2001.

[20] W. Ogata and K. Kurosawa, Optimum secret sharing scheme secure against cheating, in "Advances in Cryptology -- EUROCRYPT '96", U. Maurer, ed., *Lecture Notes in Computer Science* **1070** (1996), 200-211.
[21] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In ACM, editor, Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing: Edmonton, Alberta, Canada, August 14--16, 1989, pages 73--85. ACM Press, 1989.
[22] E.F. Brickell and D.R. Stinson. The detection of cheaters in threshold schemes. In S. Goldwasser, editor, Advances in Cryptology - CRYPTO'88, LNCS No. 403, pages 564–577. Springer-Verlag, 1988.
[23] H. Ghodosi, J. Pieprzyk, R. Safavi-Naini, and H. Wang. On construction of cumulative secret sharing. In C. Boyd and E. Dawson, editor, In Proceedings of the Third Australasian Conference on Information Security and Privacy (ACISP'98),LNCS No. 1438, pages 379–390. Springer-Verlag, 1998.
[24] H. Lin, and L. Haen. A generalised secret sharing scheme with cheater detection. In H. Imai, R. Rivest, and T. Matsumoto, editor, In Proceedings of ASIACRYPT'91, LNCS No. 739, pages 149-158. Springer-Verlag, 1993.
[25] Josef Pieprzyk, Xian-Mo Zhang: Cheating Prevention in Linear Secret Sharing. ACISP 2002: 121-135