Some Attacks on Shamir's Secret Sharing Scheme by Inside Adversaries

Deneisha Gayle Tieng [1,*] and Ederlina Nocon[1]
[1] *Mathematics Department, De La Salle University*
**Corresponding Author: deneisha_tieng@dlsu.edu.ph*

Abstract: Secret Sharing Schemes are schemes which aim to distribute a secret to a group of participants such that certain conditions must be met for them to be able to solve for the secret. One of the first secret sharing schemes which was invented was Shamir's Secret Sharing Scheme or SSSS, which is a threshold scheme that uses the concept of polynomial interpolation. The shares in a secret sharing scheme can be revealed synchronously or asynchronously. When shares are revealed synchronously, participants reveal their shares at the same time. When shares are revealed asynchronously, participants reveal their shares one at a time. This scheme was believed to be secure, until Tompa and Woll (1989) showed a way in which an inside adversary can cheat if shares are revealed asynchronously. Harn, Lin and Li (2015) also mentioned that any secret sharing schemes can be cheated when a single insider adversary reveals his/her share last in asynchronous sharing. No analysis however on synchronous sharing was made. In addition, attacks involving collaboration among inside adversaries were not yet analyzed. This paper aims to perform analysis on these cases by checking whether SSSS is prone to certain attacks made by inside adversaries. The attacks are based from the attacks on Harn and Lin (2009) and certain adjustments were made. In our results, we were able to show that SSSS can be cheated when there is one inside adversary or when there is one group of inside adversaries.

Key Words: Threshold Schemes; Lagrangian Interpolation Formula

1. INTRODUCTION

Secrets have been important to people ever since humans started to interact with each other. There are informations which we do not want other people to know, while there are other informations which we could not risk landing in the wrong hands. One example perhaps would be our ATM passwords.

A secret may be kept to oneself or kept by a group. An example of this would be a secret recipe. It is possible that only the chef knows the secret recipe, or the entire kitchen crew knows about it, but since the recipe is secret, then no one else should know about it, otherwise, any competitor may simply replicate it.

Liu (1968) posed a question:
"Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?"

Shamir (1979) stated that 462 locks and 252 keys per scientist would be necessary for this, and that the number of locks and keys will increase dramatically as the number of scientists increases. This led him to propose his threshold scheme, more popularly known as the Shamir Threshold Scheme or Shamir's Secret

Sharing Scheme (SSSS). This scheme enables a group of n people to share a secret such that at least t of them are needed to reconstruct the secret. Tompa and Woll (1989) however showed an attack on SSSS when shares are revealed asynchronously and there is one cheater. Harn, Lin, and Li (2015) also mentioned that in asynchronous secret sharing, when other shareholders are honest, a cheater can succeed in cheating by releasing his or her share last.

After the discovery of Tompa and Woll, most researches are gearing towards cheater detection, identification, and cheat immune schemes, but no further work on the cryptanalysis of SSSS was done. There was also no study on how one can cheat SSSS under synchronous sharing. The goal of this paper perform a more detailed cryptanalysis of SSSS by analyzing its susceptibility to single inside adversaries as well as against a group of inside adversaries under synchronous and asynchronous secret sharing.

Since there are different ways that researchers interpret SSSS, the paper will study the version which was taken from Stinson (2006), and we will analyze the scheme under a single group of cheaters performing four kinds of attack. Cryptanalysis of SSSS is critical because there are several new secret sharing schemes which are based from SSSS. In addition, some weaknesses of simpler schemes might have gone undetected, and thus causing these weaknesses to possibly carry on the more advanced schemes.

For the discussion of the main results, the development of the study is done as follows: (1) a review of the classic SSSS, (2) description of various attacks as described by various authors, (3)the authors' own version of attacks including the analysis of each, and (4) statement of conclusions and recommendations derived from the cryptanalysis made.

2. SHAMIR'S SECRET SHARING SCHEME

In this section, we will review Shamir's Secret Sharing Scheme as discussed in Stinson (2006) and Trappe (2002).

Definition 2.1. Let t, n be positive integers where $t \leq n$. A (t,n) −threshold scheme is a method of sharing a secret s among a set of n participants (denoted by $\mathscr{P}$), in such a way that any t participants can compute the value of s, but no group of $t-1$ participants can do so.

The notations in this paper are as follows:

The set $\mathscr{P} = \{P_i, 1 \leq i \leq n\}$ is the set of n participants or shareholders.
S is the domain of secrets or the set of all possible secrets.
T is the domain of shares or the set of all possible shares.
B is the set of participants who pool their shares together to find s.

Threshold schemes work as follows:
1. The dealer, which we denote by D, chooses the value of s.
2. When D wishes to distribute the secret s to the participants in $\mathscr{P}$, he gives each of them a part of the information which we call the share.The share is distributed secretly.
3. When a subset $B \subseteq P$ will decode the secret s, they would pool in together their shares to compute s. (It is possible that they give their shares to a trusted authority which will compute the secret for them).If $|B| \geq t$ they would be able to compute s, otherwise, they should not be able to do so.

Back in 1979, Shamir and Blakley independently introduced their own secret sharing schemes. One may refer to Shamir (1979) for the original construction. Since the original paper did not state which values are public and private, and since it was not clearly stated as to what the possible values of the secrets is, we will use the interpretation of SSSS seen in Stinson (2006). The algorithm of the Shamir (t,n)Threshold Scheme is as follows:
Let $S = Z_p$, where $p \geq n+1$ is prime. Let $T = Z_p$. In this scheme, both the secret and the shares are elements of Z_p.

Initialization Phase
1. D chooses n distinct non-zero elements of Z_p, denoted x_i, $1 \leq i \leq n$. For $1 \leq i \leq n$, D gives the values x_i to P_i. The values x_i are public.

Share Distribution
2. Suppose D wants to share a secret $s \in Z_p$. D secretly chooses $t-1$ elements of Z_p at random, which are denoted by $a_1, \ldots, a_{t-1}$.
3. For $1 \leq i \leq n$, D computes $y_i = a(x_i)$ where

$$a(x) = s + \sum_{j=1}^{t-1} a_j x^j \ mod\ p.$$

4. For $1 \leq i \leq n$, D gives the share y_i to P_i.

In this scheme, the dealer constructs a random polynomial $a(x) \in Z_p[x]$ in Step 3. This random polynomial will have a degree of at most $t-1$ and the secret is the constant term.

We now note the following proposition.

Proposition 2.2. (Stinson, 2006) In the Shamir (t, n) Threshold Scheme, any $t-1$ participants will not be able to solve for the secret s.

3. Some Attacks on Shamir's Secret Sharing Scheme

It is but inevitable that some schemes are subject to cheaters, with the motivation of fooling honest participants and keeping the secret to themselves for whatever motive they have. Tompa and Woll (1989) discussed in their paper entitled "How to Share a Secret with Cheaters" that it is possible to cheat Shamir's Secret Sharing Scheme. We will study the susceptibility of SSSS against four attacks which we will introduce later on. In our discussion, we say that a person or a group is able to succeed in cheating SSSS when they are able to achieve the following goals:

1. The secret that is solved by the group is not the true secret.
2. The cheater/s can solve for the true secret.

Harn and Lin (2009) discussed that there are three types of attack that can be performed on a secret sharing scheme:

1. **Type 1 attack**: The cheaters of this type attack can be either honest shareholders who present their shares in error accidentally or dishonest shareholders who present their faked shares without any collaboration. Each faked share of this attack is just a random integer and is completely independent with other shares.
2. **Type 2 attack:** the cheaters of this type attack are dishonest shareholders who modify their shares on purpose to fool honest shareholders. In this type attack, we assume that all shareholders release their shares synchronously. Thus, cheaters can only collaborate among themselves to figure out their faked shares before secret reconstruction; but cannot modify their shares after knowing honest shareholder's shares (i.e. we assume that all shares must be revealed simultaneously). Under this assumption, only when the number of cheaters is larger than or equal to the threshold value t, the cheaters can implement an attack successfully to fool honest shareholders.
3. **Type 3 attack:** the cheaters of this type attack are dishonest shareholders who modify their shares on purpose to fool honest shareholders. In this type attack, we assume that all shareholders release their shares asynchronously. Since shareholders release their shares one at a time, the optimum choice for cheaters is to release their shares after all honest shareholders releasing their shares. The cheaters can modify their shares accordingly.

Since the properties of the attacks of Harn and Lin (2009) were based from their construction and not from the original SSSS scheme, we will redefine these attacks omitting their claims on the properties of these attacks. In addition, since the attack of Tompa and Woll (1989) was not captured in any of the attacks in Harn and Lin (2009), we now redefine our own attacks as follows:

1. **Type A attack**: This type of attack is without collaboration. This can be done either by accident by an honest participant, or intentionally done by a cheater. Shares are released synchronously.
2. **Type B attack**: This type of attack is without collaboration. This can be done either by accident by an honest participant, or intentionally done by a cheater. Shares are released asynchronously. In this case, the attacker can make some computations in order for the secret to be some $s' \not\equiv s \ mod\ p$.
3. **Type C attack**: This type of attack involves collaborations among cheaters. The motive of this type of attack is to fool honest participants. In this type of attack, the participants disclose their shares synchronously.
4. **Type D attack**: This type also involves collaboration among cheaters, and the motive is also to fool honest participants. The shares are disclosed asynchronously.

We will examine when a cheater or a group of cheaters will be successful or not in knowing the true secret without the need for t participants to solve for it. In the succeeding discussion, we suppose that $P_{i_1}, P_{i_2}, \ldots, P_{i_m}$(where $m \geq t$) pool their shares together. Note that the assignment of P_{i_j} is arbitrary.

To aid in the cryptanalysis, we first prove the following:

Lemma 3.1. A polynomial of degree m that passes through the points $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y_{i_m})$ cannot have the same constant term as a polynomial of degrees m that passes through the points $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y'_{i_m})$ where $y_{i_m} \not\equiv y'_{i_m} \, mod \, p$.

Proof. Let $X = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y_{i_m})\}$ and let $Y = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y'_{i_m})\}$. Observe that all the x-coordinates of the elements of set X are the same as the coordinates of the elements in set Y. Thus, the values of $b_1, b_2, \ldots, b_m$ will be the same for both polynomials. We now proceed by contradiction. Suppose that the constant term of the polynomial with degree m passing through the points in X are the same as the polynomial of degree m passing through the points in Y. Applying Lagrange's Interpolation Formula will give us

$$\sum_{j=1}^{m} b_j y_{i_j} \equiv \sum_{j=1}^{m-1} b_j y_{i_j} + b_m y'_{i_m} \, mod \, p \equiv s \, mod \, p$$
$$b_m y_{i_m} \equiv b_m y'_{i_m} \, mod \, p$$
$$y_{i_m} \equiv y'_{i_m} \, mod \, p$$

Since we assumed that $y_{i_m} \not\equiv y'_{i_m} \, mod \, p$, this is a contradiction. Thus, their constant terms cannot be the same.

This lemma tells us that when exactly one participant cheats, the secret will be invalid. In addition, it does not matter what the fake secret is. As long as the share that is revealed is not the original and there is only one attacker, then the resulting secret is s' where $s' \not\equiv s \, mod \, p$. Upon knowing the other $t-1$ shares, adding his/her real share will allow him/her to solve for s, thus, being able to cheat successfully. It does not matter as well whether the attacker chose a secret at random, or made the secret equal to some chosen $s' \not\equiv s \, mod \, p$.

Remark 3.2. When one person performs Type A or Type B attack, the resulting secret is fake.

To show this, we suppose that P_{i_m} attacked. When one person performs Type A or Type B attack, the secret obtained is the constant term of the polynomial passing through points $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y'_{i_m})$, which we know is different from the constant term of the polynomial passing through points $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_m}, y_{i_m})$ where $s' \not\equiv s \, mod \, p$ by Lemma 3.1. Thus, the resulting secret is fake.

Lemma 3.3. When one person performs Type A or Type B attack, he/she can solve for the true secret s, and resulting secret $s' \not\equiv s \, mod \, p$, which is retrieved by the group is fake.

Proof. This lemma follows from Remark 3.2.

Illustration 3.4.
Refer to Example 2.1. Suppose that P_1 decides to cheat and declares 5 as his share. The secret s' will be computed as follows
$s' \equiv 5(5) + 8(4) + 1(2) \, mod \, 13 \equiv 8 \, mod \, 13$
Note that the true secret is 10. Observe that $8 \not\equiv 10 \, mod \, 13$, and thus, when the participants pool together their shares (P_1 uses a faked share 5), the computed secret s' would be 8 and this is the secret that P_2 and P_4 would know and assume to be true. Since P1 now knows the shares of the other participants, he/she can solve for the true secret s by using their shares and his/her true share as follows:
$s \equiv 5(3) + 8(4) + 1(2) \, mod \, 13 \equiv 10 \, mod \, 13$
The secret computed by P_1 this time is the true secret s. Thus, P_1 was able to solve for the true secret.

We now show the results for Type C attacks.

Lemma 3.5. When exactly one group uses Type C attack, and when $\sum_{j=1}^{k} b_j y_{i_j} \equiv \sum_{j=1}^{k} b_j y'_{i_j} \, mod \, p$ where $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ are cheaters that form a group, then the resulting secret is the true secret.
Proof. Suppose $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ are the cheaters that form a group and suppose
they contribute shares y'_{i_j} where $1 \leq j \leq k$, and where $\sum_{j=1}^{k} b_j y_{i_j} \equiv \sum_{j=1}^{k} b_j y'_{i_j} \, mod \, p$.
Then the secret is computed as $\sum_{j=1}^{k} b_j y'_{i_j} + \sum_{j=k+1}^{m} b_j y_{i_j} \equiv s' \, mod \, p$. Since $\sum_{j=1}^{k} b_j y_{i_j} \equiv \sum_{j=1}^{k} b_j y'_{i_j} \, mod \, p$ and we know that $\sum_{j=1}^{k} b_j y_{i_j} + \sum_{j=k+1}^{m} b_j y_{i_j} \equiv s \, mod \, p$, then $s \equiv s' mod \, p$.

Thus, the resulting secret is still the true secret, and the cheaters fail to deceive the other participants.

Illustration 3.6. Refer to Example 2.1. Suppose that P_1 and P_3 form a group and decide to cheat the other participants. If P_1 and P_3 declared their shares to be 1 and 2 respectively, observe that
$5(3) + 8(4) \equiv 8 \bmod 13$
and
$5(1) + 8(2) \equiv 8 \bmod 13$.
Thus, $b_1 y_1 + b_3 y_3 \equiv b_1 y_1' + b_3 y_3' \bmod 13$. Now, solving for the secret using the faked shares of P_1 and P_3 will give us
$s \equiv 5(1) + 8(2) + 1(2) \bmod 13 \equiv 10 \bmod 13$
The secret which was solved was actually the true secret, despite P_1 and P_3 submitting faked shares.

Lemma 3.7. When exactly one group uses Type C attack and if their cheated shares produce a fake secret, the group can solve for the secret.

Proof. Suppose that $P_{i_1}, P_{i_2}, \ldots, P_{i_k}$ form a group and contribute shares y'_{i_j} where $1 \le j \le k$, and where $\sum_{j=1}^{k} b_j y_{i_j} \not\equiv \sum_{j=1}^{k} b_j y'_{i_j} \bmod p$. This must be true so that the resulting secret is not the true secret s. The members of the group know the values of the other valid shares, as well as the fake secret s' generated by contributing shares y'_{i_j} where $1 \le j \le k$.They know that
$\sum_{j=1}^{k} b_j y_{i_j} + \sum_{j=k+1}^{m} b_j y_{i_j} \equiv s \bmod p$ (1)
and
$\sum_{j=1}^{k} b_j y'_{i_j} + \sum_{j=k+1}^{m} b_j y_{i_j} \equiv s' \bmod p$ (2)

Using (1) and (2), they can solve the true secret using

$$s \equiv s' - \sum_{j=1}^{k} b_j y'_{i_j} + \sum_{j=k+1}^{m} b_j y_{i_j} \bmod p.$$

Since they know the values of $s', \sum_{j=1}^{k} b_j y'_{i_j}$ and $\sum_{j=1}^{k} b_j y_{i_j}$, they can solve for the secret s.

Illustration 3.8. Refer to Example 2.1. Suppose that P_1 and P_3 form a group and decided to cheat, and declare their shares to be 7 and 3 respectively. Observe that
$5(3) + 8(4) \equiv 8 \bmod 13$
and
$5(7) + 8(3) \equiv 7 \bmod 13$

Observe that $b_1 y_1 + b_3 y_3 \not\equiv b_1 y_1' + b_3 y_3' \bmod 13$. When the participants P_1, P_3 and P_4 compute the secret using their declared shares, the computed $s'$is
$s' \equiv 5(7) + 8(3) + 1(2) \bmod 13$
$\equiv 9 \bmod 13$.
Thus, the computed secret is fake. Now, since the group already knows the true shares of the other participants, they can compute s as
$s' \equiv 5(3) + 8(4) + 1(2) \bmod 13$
$\equiv 10 \bmod 13$.
Thus, the secret computed by the group is the true secret s, while P_4 thinks that 9 is the secret.

Note that the results for Type C attack do not depend on the order in which shares are revealed, but rely only on the knowledge of the other shares. Thus, the results for Type C also hold for Type D attacks

With all these results, we have the following theorem:

Theorem 3.9. Under Case 1, the following are true:

1. When exactly one participant uses Type A or Type B attack, he/she can solve for the true secret s.
2. When exactly one group uses Type C or Type D attack, they can solve
for the true secret s.

4. CONCLUSIONS

In the study of Tompa and Woll (1989), they showed that SSSS can be cheated when shares are revealed asynchronously and the cheater reveals his/her share last. We have elucidated the scenarios in which SSSS can be cheated successfully. SSSS can be cheated as long as there is only one cheater, or one group of cheaters. As seen in the proofs of the lemma, we have shown that what is critical is the knowledge of the shares of the other participants and not the order in which the shares appear. Thus, when dealing with SSSS, it does not matter whether shares are synchronous or asynchronous. This is why results for synchronous and asynchronous SSSS have the same results for cheating.

We see in this work the significant contribution of the cryptanalysis made by way of finding some weaknesses of the SSSS in the form of certain attacks as described in Section 3. If the design of the scheme can be altered in order to address this concern then this leads to a stronger and more secure sharing scheme.

For further studies, we recommend that other researchers try to find other attacks on SSSS, especially attacks of outside adversaries since the attacks considered in this paper only involved inside adversaries. We also recommend that other researchers try to perform a similar analysis on other kinds of secret sharing scheme.

5. REFERENCES

Harn, L., & Lin, C. (2009). Detection and identification of cheaters in (t; n) secret sharing scheme. Designs, Codes and Cryptography, 52 (1),15-24.

Harn, L, & Lin, C., & Li, Y. (2015). Fair secret reconstruction in (t; n) secret sharing. Journal of Information Security and Applications, 25, 1-7.

Schneier, B. (1996). Applied Cryptography. Canada: John Wiley & Sons, Inc.

Shamir, A. (1979). How to share a secret. Communications of the ACM, 22 (22), 612-613

Stinson, D. (2006). Cryptography Theory and Practice Third Edition. Tay- lor and Francis Group, LLC Tompa, M., & Woll, H. (1989). How to Share a Secret with Cheaters. Journal of Cryptology, Vol. 1, Isssue 3, pp. 133-138. Retrieved from http://www.cs.nccu.edu.tw/raylin/UndergraduateCourse/ComtenporaryCryptography/Spring2009/Tompa.pdf

Trappe, W., & Washington, L. C. (2002). Introduction to Cryptography with Coding Theory. Upper Saddle River, New Jersey:Prentice-Hall Inc.