Achieving Correctness in Fair Rational Secret Sharing

Sourya Joyee De & Asim K Pal
sjoyeede@gmail.com,asim@iimcal.ac.in

Indian Institute of Management Calcutta

12th International Conference on Cryptology and Network Security
November 20, 2013

Problem Overview

A party in a Rational Secret Sharing (RSS) protocol may prefer to mislead others by aborting early.

'Correctness' of the reconstructed secret is jeopardized even though 'fairness' is maintained.

Some parties end up believing an incorrect value to be the correct secret.

This problem arises only for non-simultaneous channels.

Research Gap

Table: Comparison of Rational Secret Reconstruction Mechanisms

RSS Protocols	Special Preferences	Channel Type	Utility-independence
Halpern & Teague ('04)		Simultaneous broadcast	No
Gordon & Katz ('06)		Simultaneous broadcast	No
Kol & Naor ('08)	$U^{TT} > U^{NF}$	Non-simultaneous broadcast	No
Asharov & Lindell ('10)	$U^{NF} \geq U^{TT}$	Non-simultaneous broadcast	U^{NF} dependent; proved impossibility of U^{NF} independence for $(2, 2)$ case.
Fuchsbauer et al. ('10)	$U^{TT} > U^{NF}$	Non-simultaneous, point-to-point, synchronous	No
Lysyanskaya & Segal ('10)	$U^{TT} > U^{NF}$	Non-simultaneous, point-to-point, synchronous	No
Proposed protocol	$U^{NF} \geq U^{TT}$	Non-simultaneous broadcast	U^{NF} independence

Shamir's Secret Sharing Scheme

Shamir's (t, n) secret sharing scheme (where $n > t$):

$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}$
Set $a_0 = s$ where s is the secret.

Share generation: Share s_i of party P_i is given by $s_i = f(i)$.
The dealer (assumed honest) gives out a signed share to each player.

Secret reconstruction:

- Each party broadcasts his share.
- If at least t shares are obtained the secret can be reconstructed.
- $t - 1$ shares cannot give the secret.

Rational Secret Sharing

Halpern & Teague (2004) introduced players who are rational instead of good or bad.

Each rational player wants to obtain the secret alone.

In Shamir's scheme, it is in Nash Equilibrium for rational players remain silent.

Utilities and Preferences

Table: Outcomes and Utilities for $(2, 2)$ rational secret reconstruction

P_1's outcome (o_1)	P_2's outcome (o_2)	P_1's Utility $U_1(o_1, o_2)$	P_2's Utility $U_2(o_1, o_2)$
$o_1 = s$	$o_2 = s$	$U_1^{TT}(U_1)$	$U_2^{TT}(U_2)$
$o_1 = \perp$	$o_2 = \perp$	$U_1^{NN}(U_1^{-})$	$U_2^{NN}(U_2^{-})$
$o_1 = s$	$o_2 = \perp$	$U_1^{TN}(U_1^{+})$	$U_2^{NT}(U_2^{--})$
$o_1 = \perp$	$o_2 = s$	$U_1^{NT}(U_1^{--})$	$U_2^{TN}(U_2^{+})$
$o_1 = \perp$	$o_2 \notin \{s, \perp\}$	$U_1^{NF}(U_1^{f})$	U_2^{FN}
$o_1 \notin \{s, \perp\}$	$o_2 = \perp$	U_1^{FN}	$U_2^{NF}(U_2^{f})$

A party P_i has one of the following preferences:

1. $\mathcal{R}_1 : U_i^{TN} > U_i^{TT} > U_i^{NN} > U_i^{FN}$ and $U_i^{NF} \geq U_i^{TT}$
2. $\mathcal{R}_2 : U_i^{TN} > U_i^{TT} > U_i^{NN} > U_i^{FN}$ and $U_i^{NF} < U_i^{TT}$

Fairness and Correctness

Fairness

A rational secret reconstruction mechanism $(\Gamma, \overrightarrow{\sigma})$ is said to be completely fair if for every arbitrary alternative strategy $\sigma_i^{'}$ followed by party P_i, $(i \in \{1,2\})$ there exists a negligible function μ in the security parameter k such that the following holds:

$$Pr[o_i(\Gamma, (\sigma_i^{'}, \sigma_{-i})) = s] \leq Pr[o_{-i}(\Gamma, (\sigma_i^{'}, \sigma_{-i})) = s] + \mu(k)$$

Correctness

A rational secret reconstruction mechanism $(\Gamma, \overrightarrow{\sigma})$ is said to be correct if for every arbitrary alternative strategy $\sigma_i^{'}$ followed by party P_i, $(i \in \{1,2\})$ there exists a negligible function μ in the security parameter k such that the following holds:

$$Pr[o_{-i}(\Gamma, (\sigma_i^{'}, \sigma_{-i})) \notin \{s, \perp\}] \leq \mu(k)$$

Our Contribution

Our $(2, 2)$ rational secret sharing protocol has the following properties:

- It addresses both preference $\mathcal{R}_1$ and $\mathcal{R}_2$.
- It is fair and correct in the non-simultaneous channel model.
- It is independent of the utility of misleading i.e. U_{NF}.
- It is in computational strict Nash equilibrium in the presence of protocol-induced auxiliary information.

Our protocol can be easily extended to the (t, n) case.

Protocol Overview

Each rational party is given a list of sub-shares of shares of the actual secret and fake shares.

In each round, each party sends the current element in its list to the other party and reconstructs a share from the sub-shares obtained.

We use a **checking share** which is a share of the original secret as a protocol-induced membership auxiliary information to check whether the shares obtained till a certain round can be used to reconstruct the correct secret.

We overcome the disadvantages of the presence of auxiliary information by using the **time-delayed encryption** scheme used by the protocol of Lysyanskaya and Segal (2010) that tolerates players with arbitrary side information.

Membership Oracle

Membership Oracle

Let s be the actual secret and one needs to check whether x is same as the actual secret or not. S is the set of all such x. Then, a membership oracle $O : S \rightarrow \{0, 1\}$ is defined as follows:

$$O_S(x) = \begin{cases} 1 & \text{if } x = s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Correct Membership Oracle

A correct membership oracle $O : S \rightarrow 0, 1$ is a membership oracle which has the following properties:

1. $Pr[O_S(x) = 1] \leq \mu(k)$ for any $x \neq s$ and
2. $Pr[O_S(x) = 0] \leq \mu(k)$ for $x = s$.

where $\mu(k)$ is a negligible function in the security parameter k.

Protocol-induced Membership Oracle

A correct membership oracle $O^{\pi}_{q,i}$ provided by the protocol π to its participant P_i, $(i = 1, 2)$ for the qth execution of π is called a protocol-induced membership oracle.

Our protocol-induced membership oracle is linked to Shamir's (1979) (t, n) threshold secret sharing scheme.

Checking Share

The value of t is unknown to a player. He wants to reconstruct a secret from r shares ($r < n$) he has gathered.

On reconstructing a secret $s_r^{'}$ from $r^{'} < r$ shares, we can write the following:

$$f_{r'}(x) = s_{r'} + a_1^{'}x + a_2^{'}x^2 + \ldots + a'_{r'-1}x^{r^{'}-1}$$

Assume that the checking share s_q is represented as $(y_q, f(y_q) mod p)$.

Claim 1. *If $f_{r'}(y_q) = f(y_q)$, then a player can definitely conclude that $s_{r'} = s$; otherwise it concludes that $s_{r'} \neq s$.*

Time-delayed Encryption

When players have auxiliary information, then in each round, a deviating player tries to decide whether the current round is the revelation round by checking the reconstructed secret with the auxiliary information.

Once the auxiliary information tells this player that the secret has been reconstructed, the player immediately quits without sending its own share. This results in unfairness as the other player cannot reconstruct the secret.

A message that has been encrypted by a time-delayed encryption (TDE) scheme can only be decrypted after a moderate amount of time has elapsed.

In TDE (Lysyanskaya & Segal, 2010) the time delay is introduced with the help of cryptographic memory bound functions.

Our Protocol: Informal Description (1/3)

Each player is given a list of sub-shares, one for the share to be reconstructed in each round.

The minimum number of rounds r required to generate enough shares so that the secret can be reconstructed is determined by the dealer randomly from a geometric distribution with parameter β.

We want β such that

$$\beta < (U^{TT} - U^{NN})/(U^{TN} - U^{NN})$$

The dealer generates shares of the secret s according to $(r, r+1)$ Shamir's secret sharing scheme.

None of the parties are aware of the value of r.

Our Protocol: Informal Description (2/3)

The dealer also does the following:

- randomly chooses one of the $r + 1$ shares as the checking share;
- generates sub-shares of each of the remaining r shares
- generates shares of d fake secrets where d is also chosen from a geometric distribution with parameter β;

The dealer is assumed to be honest and sends the sub-shares digitally signed.

In each round, players are required to send the sub-share corresponding to the current round in their lists one by one i.e. non-simultaneously.

Our Protocol: Informal Description (3/3)

The extra share (called checking share) can be used to determine correctly whether the secret is the correct one.

The checking share acts as an indicator of the revelation round.

However, the party communicating last in any round can use it to identify the actual secret and quit before the other party obtains the secret.

We solve this problem by encrypting each share with the time-delayed encryption scheme (Lysyanskaya & Segal, 2010) and then generating sub-shares from the encrypted share.

Protocol ShareGen : The Dealer's Protocol

The dealer does the following:

1. Generate $r \sim \mathcal{G}(\beta)$.
2. $K_i, K_i^{'}, F_i \leftarrow Gen(1^k), i = 1, \ldots, r$.
3. Use $(r, r+1)$ Shamir's Secret Sharing Scheme to generate r shares of s.
4. Choose s_{check} to be the 0th share among these $(r+1)$ shares. Then, s_{check} is of the form $(y_0, f(y_0))$.
5. For each share s_i, $i = 1, \ldots, r$, compute $c_i \leftarrow Enc_{K_i}(s_i)$ and set $c_i^{'} \leftarrow (c_i, K_i^{'})$.
6. For each encrypted share $c_i^{'}$, $i = 1, \ldots, r$, generate sub-shares $c_{i,j}^{'}$ $(j = 1, 2)$ such that $c_i^{'} = c_{i,1}^{'} \oplus c_{i,2}^{'}$.
7. Generate random values $c_{i,j}^{'}$ (for $i = r+1, \ldots, r+d$ and $j = 1, 2$), d is chosen according to the geometric distribution $\mathcal{G}(\beta)$.
8. Construct list $list_j$, $(j = 1, 2)$ to contain $c_{1,j}^{'}, \ldots, c_{r+d,j}^{'}$ for player P_j $(j = 1, 2)$.

Output. Distribute to each player P_j a list $list_j$, $j = 1, 2$. Also distribute the checking share s_{check} to each player.

Protocol Reconstruct: The players'protocol (1/2)

Inputs. List of sub-shares $list_j$ received by each player P_j , $j = 1, 2$ from the dealer.

Communication Phase. P_1 communicates first as follows:

1. If in the last round (except if the current round is the first one) P_1 has not received a share within the specified deadline from P_2 or if the share received is not signed properly then abort; else continue till the Processing Phase outputs the secret.
2. Send the current share from $list_1$.
3. Check for shares sent by P_2 till the specified deadline.

P_2 communicates next as follows:

1. If in the current round P_2 has not received a share from P_1 within the specified deadline or if the share received is not signed properly then abort; else continue till the Processing Phase outputs the secret.
2. Send the current share in the list $list_2$.
3. Check for shares sent by P_1 till the specified deadline.

Protocol Reconstruct: The players' protocol (2/2)

Processing Phase.

Until the sub-shares obtained from the Communication Phase is exhausted or until the secret is obtained, each P_j $(j = 1, 2)$ does the following in the ith round of the Processing Phase:

1. Reconstruct $c_i^{'}$ from $c_{i,1}^{'}$ and $c_{i,2}^{'}$.
2. Interpret $c_i^{'}$ as $(c_i, K_i^{'})$.
3. Compute $K_i \leftarrow Unseal_{F_i}(K_i^{'})$ and find $share_i = Dec_{K_i}(c_i)$.
4. If $i > 1$, reconstruct a polynomial $f_i(x)$ of degree $(i-1)$ corresponding to the shares decrypted till the ith round; else move to the first step.
5. Now, s_{check} is $(y_0, f(y_0))$. If $f_i(y_0) = f(y_0)$ then output the constant term $f_i(0)$ of this polynomial as the desired secret and quit. Otherwise, continue. If all sub-shares obtained from the communication round are exhausted and $f_i(y_0) = f(y_0)$ does not hold then output $\perp$.

Output. Either each party outputs the secret s or each party outputs $\perp$.

Future Work

Extension of our protocol for point-to-point channels.

Extension to tolerate arbitrary side information.

Application in Rational Multi-party Computation.

Acknowlegement

We are indebted to the anonymous reviewers for their numerous useful comments. We would like to thank them for their kind efforts to help us improve our work.

Thank you!