Rational Secret Sharing without Broadcast

Amjed Shareef,
Department of Computer Science and Engineering,
Indian Institute of Technology Madras, Chennai, India.
Email: *amjedshareef@gmail.com*

Abstract

We use the concept of rational secret sharing, which was initially introduced by Halpern and Teague [2], where players' preferences are that they prefer to learn the secret than not, and moreover they prefer that as few others learn the secret as possible. This paper is an attempt to introduce a rational secret sharing scheme which defers from previous RSS schemes in that this scheme does not rely on broadcast to send messages but instead uses point to point transmissions. Not only that, but the protocol will not rely on any cryptographic primitives and is coalition resilient except for when the short player colludes with a long player.

1 Introduction

1.1 Background

The notion of secret sharing was introduced by Shamir [5]. His scheme was based upon the fact that it requires m unique points in order to define a polynomial of degree $(m-1)$. According to the scheme, a dealer generates a random polynomial, f, of degree $(m-1)$ such that $f(0)=s$, where s is the secret to be shared. Then he generates n points on the polynomial and distributes the couplet $(x_i, f(x_i))$ to every player i. If any m players come together, they will be able to regenerate the polynomial via Lagrange's interpolation and hence will be able to obtain the secret. Secret sharing is in essence the ability of several parties to work together to reconstruct a secret using information about the secret available with them. Until recently, the players involved in secret sharing were only looked upon as being either totally honest or arbitrarily malicious.

With the help of game theory, we are able to view the players in a different light, which more closely resembles human behaviour. In the case of rational secret sharing, players will behave in a way which maximizes their profit. This is modelled via a utility function whose input is the set of actions of the player and the other players, and whose output is the expected gain for that player. Halpern and Teague [2] introduced this problem with their seminal paper on rational secret sharing.

1.2 Related Work and Contribution

Previous work done includes a protocol presented by Kol and Naor [4] using a simultaneous broadcast channel to share a secret among the players. The other protocol by Kol and Naor [3] solves the same problem and is also coalition resilient, but requires cryptographic primitives. The protocol by Fuchsbauer et. al. [1] solves the rational secret sharing problem, while using simultaneous point to point channels, assuming the use of cryptographic primitives. However, there is a drawback to protocols which use a broadcast channel. The problem is that it is difficult to simulate a broadcast channel among rational players because of their rational behaviour and inclinations. So then, in order to overcome these difficulties, we must look to a new method to spread the messages instead of a broadcast channel. Such a method would be to have players send messages individually to each other, in a point to point manner. This was done by Fuchsbauer et. al. [1], but they used cryptographic primitives. The use of cryptographic primitives provides quite a bit of overhead and also brings in the problem of backward induction. Hence an interesting question arises, that of whether it is possible to come up with a rational secret sharing protocol without having to simulate a broadcast channel and without using cryptographic primitives. Our main contribution is in essence an affirmative answer to this question in the form of a protocol for rational secret sharing problem, which uses simultaneous point to point channels and which does not use cryptographic primitives.

As an extension to our work, solutions to rational multiparty computation problems, using simultaneous point to point channels, can be attempted using our solution as a basis.

1.3 Assumptions on the Utilities

We assume that the player wishes to learn the secret than not learn it and he also wishyes that as few others learn it as possible. We extend this assumption to coalitions of players as well such that any coalition wishes to learn the secret and wishes that as few other players learn the secret as possible. Let α be the upper bound on the probability that a coalition $C \epsilon \mathcal{C}$ can guess the right value in advance. Let β be the upper bound on the probability that the current iteration is definitive. It is used as the parameter to the geometric distribution from which share sizes are chosen.

In order to upper bound α and β, we use two values α_0 and β_0 respectively, the computations of which will be discussed in the next section.

2 The Protocol

2.1 Establishing the Communication Links

The number of players participating in the secret construction can vary from time to time. But the minimum number of shares required to construct the secret is constant. We consider the case where all the players come together to compute the secret. In this case, every rational player does not want to send his share to all the players. Proving that the strategy, sending share to all the players, is dominant, is difficult due to the possible collusions of the players. So we adopt a different approach where every player sends his share to at most $m-1$ other players. This is communication efficient. Hence the intrinsic complexity of the solution depends on the question, is it always possible for every player to send his share to $m-1$ players and receive $m-1$ other players' shares? We answer this question via the following lemma, which says that if $n(m-1)$ is even, then every player can be in communication with the other $m-1$ players. We show such a construction by regular graphs.

Consider an undirected graph $G(V, E)$ which represents the game. Let V denote the set of players ($V = \{p_1, \ldots, p_n\}$) and E denote the sharing relationship between two players. If there is an edge between vertices p_i and p_j, that is $(p_i, p_j) \in E$, then p_i sends his share to p_j and p_j sends his share to p_i. In this way, every vertex belonging to V should have a degree of $(m-1)$ to get the secret(as every player needs $(m-1)$ other shares). Thus, the problem is reduced to that of forming an $(m-1)$-regular graph (every node has a degree $(m-1)$) with n vertices). We present such a graph construction when $n*(m-1)$ is even.

Lemma 1 *With n vertices, forming an $(m-1)$-regular graph is possible, if $n*(m-1)$ is even.*

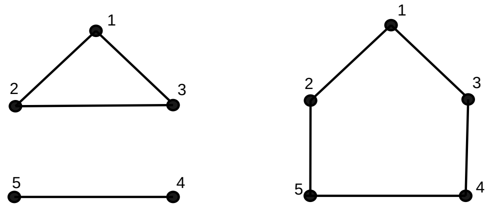
Proof: We can analyse the construction of the graph in two cases. In both the cases, we show that every vertex v_i is connected to the other $m-1$ vertices.
Case 1: $(m-1)$ is even.

$\{p_{(i+j) mod\ n};\ j = 1, 2 \ldots \frac{m-1}{2}\} \ \cup \ \{p_{(i-j) mod\ n};\ j = 1, 2 \ldots \frac{m-1}{2}\}$

Case 2: $(m-1)$ is odd and n is even
$\{p_{(i+j) mod\ n};\ j = 1, 2 \ldots \frac{m-2}{2}\} \cup \{p_{(i-j) mod\ n};\ j = 1, 2 \ldots \frac{m-2}{2}\} \cup p_{(i+\frac{n}{2}) mod\ n}$

Corollary 1 *If $n*(m-1)$ is odd, then forming an m-regular graph with n vertices is possible as $n*m$ is even.*

Suppose that $n = 5$ and $m = 3$. We cannot form a coalition of three players leaving out the other two players, as shown in fig-1. But we can form a single coalition by means of a 2-regular graph with 5 vertices as shown in fig-2, thereby ensuring that every player gets the secret.



(fig-1) (fig-2)

Figure 1: Example illustrating single coalition formation when $n = 5$, $m = 3$

2.2 Communication Set Construction Protocol

In our protocol, every player $p_i$sends his share to only a few players and likewise receives the set of shares from only a few players. A player $p_i$dynamically decides to which players he has to send his share (also receive) depending on the number of players. The *communication set* for a player p_i, CS_i, contains the identities of the players for which the player $p_i$should send his share to, and receive from also. When $n(m-1)$ is even, a player p_i's communication set, CS_i, size is of $m-1$. If $n(m-1)$ is odd, then the size of communication setof player p_i, CS_i is m, i.e. player p_i, sends his share to m other players. Note that if $n = m$, then $n * (m-1)$ is even, so it will not create any problems as $|CS_i| = m-1$.

The player $p_i$constructs the communication setas follows.

- If $n(m-1)$ is even:
  - If $(m-1)$ is even.
    1. Add players $\{p_{(i+j)\ mod\ n};\ j = 1, 2 \ldots \frac{m-1}{2}\}$
    2. Add players $\{p_{(i-j)\ mod\ n};\ j = 1, 2 \ldots \frac{m-1}{2}\}$
  - If $(m-1)$ is odd.
    1. Add players $\{p_{(i+j)\ mod\ n};\ j = 1, 2 \ldots \frac{m-2}{2}\}$
    2. Add players $\{p_{(i+j)\ mod\ n};\ j = 1, 2 \ldots \frac{m-2}{2}\}$
    3. Add player $p_{(i+\frac{n}{2})\ mod\ n}$
- If $n(m-1)$ is odd:
  1. Add players $\{p_{(i+j)\ mod\ n};\ j = 1, 2 \ldots \frac{m}{2}\}$
  2. Add players $\{p_{(i-j)\ mod\ n};\ j = 1, 2 \ldots \frac{m}{2}\}$

Table 1: Player p_i's Communication Set Construction protocol

2.3 Calculating α_0 and β_0

2.3.1 Calculating α_0

The derivation of the value of α_0 is the same as that in the paper by Kol and Naor [3].
$\alpha_0 = \alpha_0^C = min_{i\epsilon C}\{\frac{U_i - U_i^-}{U_i^+ - U_i^-}\}$

2.3.2 Calculating β_0

As with α_0, β_0 can be taken from Kol and Naor's paper [3].
$\beta_0 = \beta_0^C = min_{C\epsilon\mathcal{C}}\{min_{i\epsilon C}\{\frac{U_i - U_i^{guess,C}}{U_i^+ - U_i^{guess,C}}\}\}$

2.4 The Dealer's Share Assignment Algorithm

Each player is assigned a share which decides what values he will broadcast. The size of the share is determined by the number of cells. Out of n shares, $n-1$ of them are of size $L = l + d - 1$ and one is of size $l-1$, where l and d are chosen using $\mathcal{G}(\beta)$. Each share consists of several cells which, in turn, consist of stages, a short mask, a long mask, a masked secret, a boolean indicator and authentication information.

Dealer(y,β)

$G(\beta)$ is a geometric distribution with parameter β. Let $\mathbb{F} = GF(p)$ for $p \geq |Y|$ and the element of the secret is identified with an element of $\mathbb{F}$.

- **Create the list of possible secrets**
  - Choose l, d from $G(\beta)$. The two possible list lengths are $l-1$ and $L = l + d - 1$,i.e. the number of possible secrets. The number of the definitive iteration is denoted by l.
  - Fill the list L with random elements such that the l^{th} element is y.
- **Create Shares**
  The dealer creates n vectors, among which one list length is $l-1$ and the rest of the list lengths are $L = l+d-1$. Each cell $k \geq 1$, consists of the data which is used in the k^{th} iteration of reconstruction protocol. Every cell consists of the following elements:
  - **Short Mask:** We use this *short mask* set when $n(m-1)$ is even. The mask represents an m-out-of-n secret share. The share consists of randomly chosen elements of $\mathbb{F}$. This mask is used to unveil the secret in the next iteration, when $n(m-1)$ is even.
  - **Long Mask:** We use this *long mask* set when $n(m-1)$ is odd. The mask represents an $(m+1)$-out-of-n secret share. The share consists of randomly chosen elements of $\mathbb{F}$. This mask is used to unveil the secret in the next iteration, when $n(m-1)$ is odd.
  - **Masked Secret:** It is an element from $\mathbb{F}$. The mask (either short mask or long mask) is obtained by taking the interpolation of the previous round mask shares. The actual secret is obtained by summing the masked secret and the previous iteration's mask and the actual secret can be obtained (if the current iteration is definitive).
  - **Indicator:** It is an m-out-of-n secret share, after combining the secret we can get the actual boolean value which indicates whether the next iteration is definitive or not.
  - **Authentication information:** It contains a "tag" and "hash functions". The tag is used to prove the authenticity of the previous elements in the cell. With this tag other players can verify the correctness of the message you sent. Hash functions are used to verify the correctness of the messages sent by the other players, with probability at least $1-\beta$.

  The cell '0', added at the beginning of the vector, constitutes an m-out-of-n Shamir share of mask, and $(m+1)$-out-of-n Shamir share of mask, which are going to be used in the first iteration, and authentication information for it and to check other players' values.
- **Assign shares**
  Randomly assign shares to all the players.

Table 2: The dealer's share assignment algorithm

2.5 Secret Reconstruction Protocol

We present the player p_i's reconstruction protocol. This is similar to Kol and Naor's protocol [4]. The changes we made in the algorithm are emphasised in bold. Every player $p_i$sends his share to the set CS_i that he constructed during the *Communication Set Construction protocol.*

2.6 Theorem

Theorem 1 *Let $2 \leq m \leq n$, Y be a finite set of secrets, and dealer be an algorithm assigning m-out-of-n shares. Assume that $\alpha < \alpha_0$ and $\beta < \beta_0$. The protocol is a rational m-out-of-n secret sharing scheme for Y with running time $O(1/\beta^2)$ and number of iterations $O(1/\beta)$.*

Player$_i$(share)
Set secret_revealed ← FALSE and Cheater_detected← FALSE
Repeat until secret_revealed←TRUE **or** Cheater_detected←TRUE

- **If your share ended:**
  - Keep silent.
  - If someone has **sent share**, secret_revealed ← TRUE.
- **If your share did not end:** use the corresponding cell of share to check whether this is the last stage of this iteration.
  - If this is not the last stage:
    - Keep silent.
    - If someone has **sent share**, secret_revealed ← TRUE.
  - If this is the last stage:
    - **Send the the masked secret to** CS_i, tag, and shares of the random mask (short mask if $n*(m-1)$ is even, and long mask if $n*(m-1)$ is odd) and indicator, as they appear in the corresponding cell of share.
    - If more than a single player did not **send the share**, or if some of the messages do not pass the authenticity check (the tags and hash functions do not match), cheater_detected ← TRUE.
    - **If Out of** CS_i**, all but a single player send share**, or if the reconstructed indicator shows that the iteration is definitive, secret_revealed ← TRUE. **Let the Masked secret be** MS
- **Leave the game:** Quit and output the current possible secret (obtained by subtracting the mask reconstructed using the shares **received** in the previous iteration **from the Masked secret,** MS**, constructed after definitive iteration).**

Table 3: Player p_i's reconstruction protocol

Proof: If everyone follows the protocol, then with probability 1 they will get the secret. Since the probability that a given iteration is the definitive iteration is β, the number of iterations required to get the secret is $O(1/\beta)$.
Given that each iteration consists of several stages, also determined by the parameter β, we can easily see that the running time of the protocol is $O(1/\beta^2)$.
In order for this protocol to be a rational m-out-of-n secret sharing scheme, players should have an incentive to not deviate. This incentive can be created via a function of their utilities. The idea is that the players should gain more from following the protocol than from deviating. The same holds true for coalitions. The utility of a player i when following the protocol is U_i. Let U_i^+ be the utility of a player who successfully guesses the secret. Let $U_i^{guess,C}$ be the utility of a player i belonging to coalition C when the coalition does not participate in the protocol and instead tries to guess the secret.
We can provide an incentive for players to follow the protocol so long as the following inequality holds true for every player i, belonging to a coalition C.

$$\beta.U_i^+ + (1-\beta).U_i^{guess,C} < U_i$$
$$\beta.(U_i^+ - U_i^{guess,C}) < U_i - U_i^{guess,C}$$
$$\beta < \frac{U_i - U_i^{guess,C}}{U_i^+ - U_i^{guess,C}}$$

As we can see, it suffices to require $\beta < \beta_0$ for $\beta_0 = min_{C\epsilon\mathcal{C}}\{min_{i\epsilon C}\{\frac{U_i - U_i^{guess,C}}{U_i^+ - U_i^{guess,C}}\}\}$ where $\mathcal{C}_{m-1}$ is the set of coalitions of size at most $m-1$.

3 Conclusion and Open Problems

We have successfully presented the protocol for rational secret sharing with simultaneous point to point channels, without using cryptographic primitives. It shows a way to come up with the solutions to the rational multiparty computation problems, which are information theoretically secure. Our protocol is collusion free, except when the short player colludes with any long player. An interesting question that arises is if it is possible to find a solution to rational secret sharing using point to point non-simultaneous channels.

Acknowledgement: The authors would like to acknowledge William Kumar Moses, Jr., for his insightful remarks on theorem.

References

[1] Georg Fuchsbauer, Jonathan Katz, and David Naccache. Efficient rational secret sharing in standard communication networks. To appear in TCC '10.

[2] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation: extended abstract. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 623–632, New York, NY, USA, 2004. ACM.

[3] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *TCC '08*, pages 320–339, 2008.

[4] Gillat Kol and Moni Naor. Games for exchanging information. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 423–432, New York, NY, USA, 2008. ACM.

[5] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, 1979.