The Deterministic Protocol for Rational Secret Sharing

Maleka S
Indian Institute of Technology Madras
maleka.smile@gmail.com

Amjed Shareef
Indian Institute of Technology Madras
amjedshareef@gmail.com

C. Pandu Rangan[1]
Indian Institute of Technology Madras
rangan@cs.iitm.ernet.in

Abstract

We consider the rational secret sharing problem introduced by Halpern and Teague[3], where players prefer to get the secret than not to get the secret and with lower preference, prefer that as few of the other players get the secret. The impossibility of a deterministic protocol for rational secret sharing is proved by Halpern and Teague[3]. The impossibility result is based on the fact that a rational player always chooses a dominating strategy and so there is no incentive for a player to send his secret share. This rational behavior makes secret sharing impossible, but there is an interesting way by which we can force rational players to cooperate for achieving successful secret sharing. A rational player may be deterred from exploiting his short term advantage by the threat of punishment that reduces his long term payoff. This can be captured by the repeated interaction of players. Hence, we study rational secret sharing in a scenario, where players interact repeatedly in several rounds which enables the possibility of secret sharing among rational players. In our model, the dealer, instead of sending shares, forms polynomials of the secret shares and sends points on that polynomial (say subshares) to the players. The dealer constructs polynomials in a manner that the degrees of polynomials used differ by at most one and each player is not aware of the degree of polynomial employed for others. The players distribute shares in terms of subshares. We show a surprising result on the deterministic protocol for rational secret sharing problem in synchronous model. This is the first protocol that achieves rational secret sharing in a reasonable model to the best of our knowledge.

1. Introduction

Secret sharing is a widely known primitive in modern cryptography. More formally, in a secret sharing scheme there is a unique player called $dealer$ (player 0) who wants to share a secret s among n players, $p_1, \ldots, p_n$. The dealer sends every player a share of the secret in a way that any group of m (threshold value) or more players can together reconstruct the secret but no group of fewer than m players can.

Shamir's Secret Sharing Scheme[6] is based on the fact that, it takes m points to define uniquely a polynomial of degree $(m-1)$. The idea is that the dealer who shares the secret among the players, chooses a random $(m-1)$ degree polynomial f, such that $f(0) = s$, and sends the shares to the players such that every player p_i, $i = 1, \ldots, n$ receives the share $f(i)$. Any m players can recover the secret by reconstructing the polynomial through Lagrange's Interpolation. Any subset of players of size less than m cannot reconstruct the polynomial (even if they have infinite computing power).

1.1. Game Theory in Secret Sharing

Game theory provides a clean and effective tool to study and analyze the situations where decision-makers interact in a competitive manner. Game theoretic reasoning takes into account, which strategy is the best for a player with respect to every other player's strategy. Thus, the goal is to find a solution that is the best for all the players in the game. Every player's decision is based on the decision of every other player in the game and hence, it is possible to reach the equilibrium state corresponding to the global optima.

In distributed computing or secret sharing or multi-party computation, the players are mostly perceived as either honest or malicious players. Honest player follows the protocol perfectly whereas the malicious player behaves in an arbi-

[1]Work supported by Foundation Research in Cryptology. Sponsored by Microsoft Research, INDIA.

978-1-4244-1694-3/08/$25.00 ©2008 IEEE

trary manner. Halpern and Teague[3] introduced the problem of *secret sharing* assuming that the players are rational, which is known as *rational secret sharing*. In rational secret sharing, player's behavior is selfish. They have their own preferences and utility function (the profit they get). They always try to maximize their profits and behave accordingly.

For any player p_i, let w_1, w_2, w_3, w_4 be the payoffs obtained in the following scenarios.

w_1 – p_i gets the secret, others do not get the secret
w_2 – p_i gets the secret, others get the secret
w_3 – p_i does not get the secret, others do not get the secret
w_4 – p_i does not get the secret, others get the secret

The preferences of p_i is specified by $w_1 > w_2 > w_3 > w_4$. In brief, every player primarily prefers to get the secret than to not get it and secondarily, prefers that the fewer of the other players that get it, the better. The least preferred scenario for p_i is the situation, where he does not get the secret and others get it. A rational player follows the protocol only if it increases his expected utility.

1.2. Related Work

Consider any arbitrary player, say p_i. He needs $(m-1)$ shares from others to compute the secret. If other players (at least $(m-1)$) send him their shares, then he gets the secret, otherwise he cannot. This does not depend on whether he sends his share to others or not, as all the players are assumed to send their shares simultaneously. So, there is no incentive for any player to send his share. Reasoning in a similar way, no player might send his share. This impossibility result is proved by Halpern and Teague[3]. They show that rational secret sharing is not possible with any mechanism that has a fixed running time by iterated deletion of weakly dominated strategies (the strategy of not sending the share weakly dominates the strategy of sending the share). They also proposed a randomized protocol for $n \geq 3$. All these results apply to multi-party computation. Gordon and katz[2] improved the original protocol and additionally they proposed a protocol for $n = 2$ for rational secret sharing and rational multi-party computation. Abraham *et.al.* [1] analyzed rational secret sharing and rational multi-party computation in a extended setting where players can form coalitions. They use a trusted third party as mediator. Lysyanskaya and Triandopoulos[4] analyzed multi-party computation in mixed behavior model, where players are rational or malicious using a trusted mediator. The malicious adversary can control at most $(\lceil n/2 \rceil - 2)$ players.

1.3. Intuition and Contribution

The basic intuition is that, repeated interaction allows socially beneficial outcomes. In other words, if interaction is repeated, then socially beneficial outcomes that cannot be obtained by players with short-term objectives can be obtained by players with long-term objectives. In our model, we create an environment for the players to interact repeatedly. In brief, the dealer, instead of sending shares, forms polynomial for every secret share and sends points on that polynomial (say subshares). Now, to obtain other player's secret share, a player has to interact with him repeatedly. Only after receiving all his subshares, a player can construct his secret share through Lagrange's interpolation. This repeated interaction of players enables secret sharing and is explained in Section 3 in detail.

The major contribution of our work is that we enable the possibility of secret sharing among the rational players. This is the first deterministic protocol for rational secret sharing to the best of our knowledge.

1.4. Model and Assumptions

Let S be the secret, which the dealer wants to share with n players. Let $\{p_1, \ldots, p_n\}$ be the set of n rational players and m be the threshold of the shares to obtain the secret S. The dealer constructs a random polynomial $F(x)$ of degree $(m-1)$ and computes the values $\{F(1), \ldots, F(n)\}$. The points $\{(1, F(1)), \ldots, (n, F(n))\}$ lie on the polynomial and the polynomial $F(x)$ can be uniquely constructed if any m of them are obtained. The values $\{F(1), \ldots, F(n)\}$ are generally referred as secret shares and are denoted by $\{s_1, \ldots, s_n\}$. In our model, the dealer constructs random polynomials $f_1, \ldots, f_n$ of degree $d_1, \ldots, d_n$ for the secret shares $s_1, \ldots, s_n$ respectively. The polynomials are constructed in a manner that, given any two polynomials f_i, f_j with degree d_i, d_j, then $|d_i - d_j| \leq 1$ where $i \neq j$. For every secret share s_i, the dealer computes $\{f_i(1), \ldots, f_i(d_i + 1)\}$, which are known as subshares and are denoted by $\{s_{i1}, \ldots, s_{i(d_i+1)}\}$ and sends them to the player p_i.

We assume that all players are connected to each other through secure private channels independently, which ensures that a player can send his share to a selected number of players. The underlying network is synchronous, means all the players are synchronized with respect to a global clock. Hence, all the players start and end the game at the same time. The messages will be delivered in fixed amount of time and the communication is guaranteed. The subshares are authenticated by the dealer, and therefore a player cannot send incorrect value as a subshare to other players. All players are assumed to be computationally bounded. There is no trusted mediator and the dealer is assumed to be honest. Players are rational, patient enough and care for their future payoff. We model the secret sharing as a game, denoted by Γ.

1.5. Paper Outline

In the next section, we briefly explain the basics of Game Theory. Section 3 presents the two player protocol for the RSS game. In section 4, we extend these results to n players and propose the n player protocol. Finally, Section 5 concludes the paper and gives an insight on open problems in further direction.

2. Basics of Game Theory

We define some basic terminology of game theory in this section [5].
A *strategy* can be defined as a complete algorithm for playing the game, implicitly listing all moves and countermoves for every possible situation throughout the game. And a *strategy profile* is a set of strategies for each player which fully specifies all actions in a game. A strategy profile must include one and only one strategy for every player.

Let $G(N, L, U)$ represents an n persons game, where N is a finite set of n players $(p_1, \ldots p_n)$, $L = \{L_1, \ldots, L_n\}$ is a set of actions for each player p_i, $i \in \{1, \ldots, n\}$ and $U = \{u_1, \ldots, u_n\}$ is a utility function for each player, where $u_i : L->$R.

2.1 Repeated Games

Repeated games capture the idea that a player can condition his future game's move based on the previous game's outcome. In repeated games, the players interact several number of times $(\Gamma_1, \Gamma_2, \cdots)$. We assume that the players make their moves simultaneously in each game. The set of the past moves of all the players is commonly referred to as the history H of the game. History is uniquely defined at the beginning of each game ($h_1, h_2, \ldots$ and $h_1 = 0$) and the future move depends on the history. In repeated games, the users typically want to maximize their payoff for all the game they play. Hence, every player p_i tries to maximize his payoff function u_i. In some cases, the objective of the player can be to maximize their payoff only for the current game (which is equivalent to a game, which is played only once). Such game is known as short-sighted game. If the players try to maximize their payoff throughout the repeated game, then it is a long-sighted game. If the game is played finite number of times, then it is a finite repeated game. Otherwise, it is an infinite repeated game.

3. The Protocol

We denote our game by $\Gamma(n, m)$, where n is the number of players participating in the game and m is the threshold value of the number of shares to obtain the secret. Every player has two actions namely, sending the subshare and not sending the subshare. Let us denote the action of sending the subshare by 'C' and not sending by 'D'. Then, the strategy of a player for always not sending is $\{D, D, \ldots\}$ and for always sending is $\{C, C, \ldots\}$. In every round, the strategy profile (strategies chosen by all the players), is denoted by n-tuple $(c_1, c_2, \ldots, c_n)$, where $c_i = C$ or $D, i \in \{1, \ldots, n\}$.

A rational player chooses a strategy that gives him the maximum payoff. For every round, a player has two actions (sending and not sending) and he chooses the one which gives him the long term benefit. Players interact repeatedly in every round and have the threat of not receiving further subshares (as others follow grim trigger strategy) if they do not send the subshare in current round. The strategy of sending the subshare, C dominates the strategy of not sending the subshare, D. Hence, every player chooses the strategy C. Thus, the repeated interaction of the players is creating an incentive for the players to send their subshares. The Rational Secret Sharing (RSS) is similar to the Repeated prisoners' dilemma in many aspects for our protocol. We first discuss the strategy of the players, then analyze the two player protocol. In next section, we describe the generalized protocol for n players.

3.1. Punishment Strategy : Grim Trigger Strategy

1. choose C as long as the other players choose C.
2. In any game some player chooses not sending (i.e., chooses D), then choose D in every subsequent game.

The grim trigger strategy for a player p_i in rational secret sharing game is defined as:
$s_i(\phi) = C$ (player p_i chooses C at the start of the game, ϕ denotes initial history), and

$$s_i(h_1, \ldots, h_q) = \begin{cases} \text{C} & \text{if } (h_{j1}, \ldots, h_{jq}) = (C, \ldots, C) \\ & \text{for every other player } p_j, j \neq i. \\ \text{D} & \text{otherwise.} \end{cases}$$

That is, the player p_i chooses C after any history in which every previous action of every player was C, and D after any other history. In other words, a player chooses C until he gets the expected number of subshares in every round.

3.2. Two Player Protocol for Rational Secret Sharing

Consider two players, say A and B. According to the protocol, the dealer constructs a polynomial F of degree 1 for the secret S, and calculates two shares of the secret, say s_1 and s_2. Now, instead of distributing the shares s_1 and

s_2 to the players, the dealer constructs two polynomials f_1 and f_2 of degree D_1 and D_2 for the secret shares s_1 and s_2 respectively. The dealer constructs the polynomials in such a manner that D_1 and D_2 differ by a value at most 1, i.e., $|D_1 - D_2| \leq 1$. The intention behind the degree difference is to ensure that the number of subshares sent to the players by the dealer also differ by a difference of atmost one.

More clearly, let the minimum number of subshares computed by the dealer using f_1 and f_2 be d_1 and d_2 respectively. To reconstruct the polynomials f_1 and f_2 and hence the secret shares s_1 and s_2, the minimum number of subshares required are (D_1 + 1) and (D_2 + 1) respectively. Hence, $|d_1 - d_2| \leq 1$. Let us denote the subshares of the secret share s_1 by $\{s_{11}, \ldots, s_{1d_1}\}$ and s_2 by $\{s_{21}, \ldots, s_{2d_2}\}$. The dealer sends authenticated subshares of the secret shares to the players. The players are aware of the procedure followed by the dealer, but are not aware of the degree of the other player's polynomial. In other words, A is not aware of the number of subshares B has and vice versa. They just know that their degree difference will be at most 1 (the difference beetween number of subshares is 1). Now, the game starts with A and B sending their subshares to each other simultaneously. We name our protocol as *Two round Rational Secret Sharing (TRSS) protocol.*

Protocol for dealer:

1. Construct a random polynomial $F(x)$ of degree 1 for the secret S, and compute the secret shares $\{s_1, s_2\}$.
2. Construct polynomials f_1 and f_2 with degree D_1 and D_2 for the secret shares s_1 and s_2 respectively, such that the degree difference between the two polynomials is at most 1, i.e., $|D_1 - D_2| \leq 1$.
3. Use f_1 to compute subshares $\{s_{11}, \ldots, s_{1d_1}\}$ of s_1, where d_1 = (D_1 + 1) and send the authenticated subshare set to the player A. Similarly, use f_2 to compute subshares $\{s_{21}, \ldots, s_{2d_2}\}$ of s_2, where d_2 = (D_2 + 1) and send the authenticated subshare set to the player B.

Protocol for players A and B:

1. In the first round, player A sends his subshare s_{11} to player B and player B sends his subshare s_{21} to player A.
2. In every round r, $r > 1$, the players send their subshares (s_{1r} or s_{2r}) to the other player if and only if they receive the $(r-1)^{th}$ subshare ($s_{1(r-1)}$ or $s_{2(r-1)}$) of the other player.

The Rational player always prefer to obtain the secret rather than not obtaining the secret. This acts as an incentive for the rational player to send his subshare in the first round. From the next round onwards, the threat of punishment acts as an incentive to send the subshare. If a player does not send his subshare in the current round (including first), then he will not receive the other player's subshare from the next round onwards. Hence, he looses the chance of obtaining the secret. This explanation can be verified by refering the preferences and payoffs mentioned in the section 1.1.

Lemma 1: *If two rational players A and B play the RSS game $\Gamma(2,2)$, then each player definitely sends his subshare to his partner through TRSS protocol till $(d-1)^{th}$ round, where $d = min(d_1, d_2)$.*

Proof : Suppose, in k^{th} $(k < d)$ round the player A does not send his subshare s_{1k} to the player B. Then, player B chooses grim trigger strategy and henceforth never sends his subshares, $\{s_{2(k+1)}, \ldots, s_{2d_2}\}$ to A. Thus, player A cannot obtain B's subshares from $(k+1)^{th}$ round onwards and his payoff will be w_3. If the player A sends his subshare s_{1k}, then his payoff would be w_2. So, for either of the players the payoff is high if they chooses sending rather than not sending in any particular round. And every player primarily prefers to get the secret than not to get the secret. Hence, both the players choose to send their subshare till $(d-1)^{th}$ round definitely. □

3.3. How the protocol works ?

Every player keeps sending his subshare until the other player sends him his subshare. Either of the players do not know when the other player's subshares will end. But, the players are aware of the fact that the degree of the polynomials differ by at most 1 (the difference beetween number of subshares is 1). Every player follows the protocol and sends his first subshare. From lemma 1, in any given round r ($r < min(d_1, d_2)$), either of the players send their subshares iff they had received the $(r-1)^th$ round's subshare. The transferring of subshares continue for sure till either of the player's subshares are exhausted. We analyze the protocol for the 3 possible cases in d^{th} round, where $d = min(d_1, d_2)$.

1. A has less subshares than B.
2. B has less subshares than A.
3. A and B have equal number of subshares.

case 1: A has less subshares than B

Suppose player A has d_1 subshares and player B has d_2 subshares, such that $d_1 < d_2$. Therefore, the value of d in this case is d_1. Both the players are not aware of number of shares other player has. Players are not even aware of value d as they do not know the number of subshares the other

player has. From lemma 1, till $(d-1)^{th}$ round, both the players have exchanged $(d-1)$ subshares.

In d^{th} round, the player A sends his d^{th} (last) subshare, s_{1d} to the player B expecting that B might have $(d+1)^{th}$ subshare, $s_{2(d+1)}$. If he does not send d^{th} subshare, s_{1d}, then he might loose the chance of getting the subshare $s_{2(d+1)}$ and hence the secret share s_2 as well as the secret S, as player B chooses grim trigger strategy. Even player B does not know how many subshares player A has. So, if player B does not send his d^{th} (last but one) subshare, s_{2d} in d^{th} round, he thinks that he might loose the chance of obtaining palyer A's $(d+1)^{th}$ subshare, $s_{1(d+1)}$ and hence the secret share s_1 as well as the secret S, as player A chooses grim trigger strategy.

In $(d+1)^{th}$ round, as player A's subshares are exhausted he remains idle without sending any subshare to player B. Player B does not know how many subshares does the player A has. If player B does not send his $(d+1)^{th}$ subshare, $s_{2(d+1)}$ in $(d+1)^{th}$ round, he thinks that he might loose the chance of obtaining player A's $(d+2)^{th}$ subshare, $s_{1(d+2)}$ and hence the secret, as player A chooses grim trigger strategy. So, the player B sends his $(d+1)^{th}$ subshare, $s_{2(d+1)}$ to the player A. After receiving B's $(d+1)^{th}$ subshare, $s_{2(d+1)}$, A will become aware of the number of subshares B has and computes B's secret share. After $(d+1)^{th}$ round, the player B realizes that A has only d subshares and so he did not send his $(d+1)^{th}$ subshare. Thus, B also computes the secret share and gets the secret. In this way, both the players get the secret.

case 2: B has less subshares than A
It is similar to case 1, with A and B exchanging their roles.

case 3: A and B have equal number of subshares
In this case, both the players have d number of subshares. In $(d-1)^{th}$ round (last but one round), both the players expect that other player might have d subshares, hence they exchange their subshares. In d^{th} round, they exchange d^{th} subshare expecting the other player to have $(d+1)$ subshares. In $(d+1)^{th}$ round, no player sends his subshare to the other player. Hence, both the players conclude that other player's subshares are exhausted and constructs the secret share and thereby the secret S.

Lemma 2 : *If two rational players A and B play the RSS game $\Gamma(2,2)$, then each player always sends his subshare to his partner through TRSS protocol.*
Proof : Easy observation from the above explanation given in subsection 3.3. □

Theorem 1 : *If two rational players A and B play the RSS game $\Gamma(2,2)$, then secret sharing is possible and every player gets the secret through TRSS protocol.*

Proof : From lemma 2, both the players send all their subshares to each other through TRSS protocol . Hence, they construct the secret. □

3.4. Illustration of TRSS protocol through an Example

Let us consider the values of the degrees D_1 and D_2 of the polynomials f_1 and f_2 be 4 and 5 respectively. Therefore, the values of d_1 and d_2 will be 5 and 6 respectively. Hence, the player A has the subshare set $\{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and the player B has the subshare set $\{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}\}$. Every player needs all the subshares of the other player to obtain the other player's secret share and hence the secret. Player A expects player B to have either 4 or 5 or 6 subshares and player B expects player A to have either 5 or 6 or 7 subshares.

Here, the value of d will be $min(5,6)$, which is 5. From lemma 1, till the 4^{th} round both the players exchange their shares, as it is the profitable strategy. Let us analyse the profitable strategy from the 5^{th} round onwards. In the 5^{th} round, player A sends his 5^{th} subshare, s_{15} to player B expecting that player B might have either 5 or 6 subshares and that, if he does not send, player B may follow grim trigger strategy, thus loosing the chance of obtaining the secret. Similarly, player B sends his 5^{th} subshare, s_{25} to player A expecting that player A might have either 5 or 6 or 7 subshares. At the end of 5^{th} round, both will be receive other player's 5^{th} subshare. In the 6^{th} round, player A does not have any more subshares to send and hence can not send any. But, the player B sends his 6^{th} subshare, s_{26} to player A expecting that player A might have either 6 or 7 subshares. At the end of the 6^{th} round, both the players receive all the subshares of other player. Thus, both the players obtain each other's secret share and hence obtain the secret S.

4. Generalized (n player) Protocol for Rational Secret Sharing

For n players, the dealer constructs random polynomials $f_1, \ldots, f_n$ of degree $D_1, \ldots, D_n$ for the secret shares $s_1, \ldots, s_n$ respectively. The polynomials are constructed in a manner that, given any two polynomials f_i, f_j with degree D_i, D_j, then $|D_i - D_j| \leq 1$ where $i \neq j$. Let the number of subshares formed using the polynomial f_i, whose degree is D_i be d_i, where $d_i = D_i + 1$ (same explanation as given in section 3.2). Then, the difference between any two given number of subshares will be one, i.e., $|d_i - d_j| \leq 1$ where $i \neq j$.

For every secret share s_i, the dealer computes $\{f_i(1), \ldots, f_i(d_i)\}$, which are known as subshares and are denoted by $\{s_{i1}, \ldots, s_{id_i}\}$ and sends them to the player p_i,

$1 \leq i \leq n$. All the subshares sent by the dealer are authenticated and no player can send an incorrect value as his subshare to other players.

To obtain the secret S, every player needs $(m-1)$ shares of the secret S. Thus, every player needs all the subshares of at least $(m-1)$ players to get the secret. So, every player should collaborate with $(m-1)$ other players to obtain the secret. When a set of players of size m is formed, the protocol to be followed is given hereunder. We name our protocol as $NRSS$.

The generalized protocol for n players
Protocol for dealer:

1. Construct a random polynomial $F(x)$ of degree $(m-1)$ for the secret S, and compute the secret shares $\{s_1, \ldots, s_n\}$.
2. Construct polynomials $f_1, \ldots, f_n$ for secret shares $\{s_1, \ldots, s_n\}$ such that the degree difference between any two polynomials is at most 1, i.e., for any two polynomials f_i, f_j with degree D_i, D_j, $|D_i - D_j| \leq 1$ should be satisfied, where $i \neq j$. Even the corresponding number of subshares should differ by at most 1, i.e., $|d_i - d_j| \leq 1$ where $i \neq j$.
3. Use f_i to compute subshares $\{s_{i1}, \ldots, s_{id_i}\}$ of s_i and send the subshare set to the player p_i, where $1 \leq i \leq n$.

Protocol for player p_i:

1. In the first round, every player p_i sends his subshare s_{i1} to $(m-1)$ players.
2. In every round r (except for the first round), every player p_i sends his subshare $s_{ir}$to other players if and only if the subshares of other $(m-1)$ players corresponding to $(r-1)^{th}$ round have been received.

We illustrate the protocol with an example. Suppose, $\{p_1, \ldots, p_m\}$ be the set of m players who formed a group and want to construct the secret. At the beginning of the protocol, every player p_i $(i \leq m)$ has the subshare set $\{s_{i1}, \ldots, s_{i(d_i)}\}$ from which he can always obtain his share of the secret, s_i. In the first round, every player p_i $(i \leq m)$ sends his subshare s_{i1} to $(m-1)$ other players as per the protocol. The interaction between any two players is same as that of the two player protocol. From lemma 1, every player has an incentive to send his subshare and so exchanging of subshares continue. The condition, given any two polynomials f_i, f_j with degree D_i, D_j, then $|D_i - D_j| \leq 1$ where $i \neq j$, assures that a player cannot become aware of the degree employed for the other player's polynomial until he sends all his subshares. Hence, every player is forced to send all his subshares to get the other player's share of the secret. Thus, the above protocol assures successful secret sharing, if a set of m players follow the protocol.

Theorem 2: *In a RSS game $\Gamma(n, m)$, if any m players come together, then through $NRSS$ protocol it is possible to construct the secret S, where n is the number of shares and m is the threshold of shares.*
Proof : From lemma 2, if two rational players p_i and p_j are involved in the rational secret sharing game and follow the $TRSS$ protocol, then each player sends all his subshares to the other player, irrespective of the degree difference between their polynomials. Thus, both the players obtain the secret share of each other. In case of the generalized protocol $NRSS$, even though there are m players, the interaction between any two players is same as that of the $TRSS$ protocol. Therefore, by the end of the $NRSS$ protocol, each player obtains the secret share of other $(m-1)$ players. Hence, obtains the secret S. □

5. Conclusions and Open Problems

We propose a deterministic protocol for rational secret sharing by creating an environment where players need to interact repeatedly. This enables the possibility of secret sharing among the rational players. In our model, the dealer instead of sending shares to the players, forms polynomials of the secret shares and sends points (subshares) on that polynomial. Now, the players distribute subshares instead of shares, thereby they interact several times and concentrate more on long term goal enabling the possibility of secret sharing. We proposed the first deterministic protocol for rational secret sharing problem in synchronous model. These results are likely to be applicable for rational multi party computation. Protocol for asynchronous model with our model is left open. We expect that with our results extend the scope for problem solving strategies in asynchronous model and noncooperative computing problems can be enhanced. The rational secret sharing problem without authenticated shares can be attempted.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *25th ACM PODC*, pages 53–62, 2006.

[2] S. Gordon and J. Katz. Rational secret sharing, revisited. In *SCN*, volume 4116, pages 229–241, 2006.

[3] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In *36t ACM Symposium on Theory of Computing(STOC)*, pages 623–632, 2004.

[4] A. Lysyanskaya and N. Triandopoulos. Rationality and adversarial behaviour in multi-party computation (extended abstract). In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 180–197. Springer, 2006.

[5] M. Osborne. *An Introduction to Game Theory*. Oxford University Press., 2004.

[6] A. Shamir. How to share a secret. In *Communications of the ACM, 22:*, pages 612–613, 1979.