

## AN EFFICIENT AND INFORMATION THEORETICALLY SECURE RATIONAL SECRET SHARING SCHEME BASED ON SYMMETRIC BIVARIATE POLYNOMIALS

CHRISTOPHE TARTARY

*Institute for Theoretical Computer Science,  
Institute for Interdisciplinary Information Sciences, Tsinghua University,  
Beijing 100084, People's Republic of China  
ctartary@mail.tsinghua.edu.cn  
<http://itcs.tsinghua.edu.cn/~ctartary/>*

HUAXIONG WANG

*Division of Mathematical Sciences, School of Physical and Mathematical Sciences,  
Nanyang Technological University,  
21 Nanyang Link, Singapore 637371  
hxwang@ntu.edu.sg  
<http://www3.ntu.edu.sg/home/hxwang/>*

YUN ZHANG

*Division of Mathematical Sciences, School of Physical and Mathematical Sciences,  
Nanyang Technological University,  
21 Nanyang Link, Singapore 637371  
ZHAN0233@e.ntu.edu.sg*

Received 28 February 2011

Accepted 7 June 2011

Communicated by Cunsheng Ding and Qi Wang

The design of rational cryptographic protocols is a recently created research area at the intersection of cryptography and game theory. In this paper, we propose a new  $m$ -out-of- $n$  rational secret sharing scheme requiring neither the involvement of the dealer (except during the initial share distribution) nor a trusted mediator. Our protocol leads to a Nash equilibrium surviving the iterated deletion of weakly dominated strategies for  $m \geq 4$ . Our construction is information theoretically secure and it is immune against backward induction attacks. Contrary to Kol and Naor who used a specific cryptographic primitive in their TCC'08 paper (namely, meaningful/meaningless encryption), the immunity of our scheme is based on the use of bivariate polynomials and one-time pads. To the best of our knowledge, it is the first time that such polynomials have been used for rational secret sharing. Our scheme is efficient and does not require any physical assumptions such as envelopes or ballot boxes. As most of existing rational protocols, our construction requires simultaneous broadcast channels. However, our proposed scheme does not require any computational assumption and it provides information theoretical security.

*Keywords:* Secret sharing scheme; rational cryptography; information theoretical security; bivariate polynomial.

## 1. Introduction

### 1.1. Preliminary

In 1979, Shamir [1] and Blakley [2] independently introduced the concept of *secret sharing scheme* (SSS) in order to facilitate the distributed storage of private data in an unreliable environment. Since then, secret sharing has become a major building block for cryptographic primitives in particular in the area of *multiparty computation* (MPC) [3]. The goal of a (perfect) SSS is to distribute a secret value  $s$  amongst a finite set of participants  $\mathcal{P} = \{P_1, \dots, P_n\}$  in such a way that only specific subsets of  $\mathcal{P}$  can reconstruct  $s$  while the others have no information about this secret element whatsoever.

Traditional cryptographic models assume that some parties are honest (i.e. they faithfully follow a given protocol) while others are malicious participants against whom the honest players must be protected. However, in many real-world applications, a participant will choose to be dishonest if deviating from the protocol will provide him with some advantage. Game theory can be used to model such a situation where players are *self-interested* (i.e. *rational*). In this representation, each participant  $P_i$  has a utility function  $U_i$  which will dictate his strategy during the execution of the protocol. Under this new model, the important question is to design meaningful protocols. Indeed, using natural assumptions regarding the utilities of the players, a classical threshold SSS completely fails: each player is better off withholding his share no matter what the other players do and, as a result, the secret  $s$  will never be reconstructed. For similar reasons, generic MPC solutions [4, 5] are not applicable since participants are sorted into "good" and "bad" members from the beginning of the protocol.

Halpern and Teague introduced the first general approach for threshold rational secret sharing in 2004 [6]. Their paper was followed by several subsequent results [7, 8, 9]. The basic constructions presented in those papers have the disadvantage of requiring the permanent involvement of the dealer even after the initial share distribution. To overcome this drawback, [7, 8, 9] replace the dealer by several iterations of MPC protocols. Unfortunately, as pointed out by Fuchsbauer *et al.* [10], those MPC functionalities are complex and it is unclear whether this approach is computationally efficient.

Another noticeable point of the constructions mentioned above is the fact that a digital signature is used at the beginning of these rational protocols to ensure the correctness of the shares to be distributed by the dealer. Since no rational protocols can have a commonly-known bound on its running time (see [6] for details), Kol and Naor pointed out that the signature scheme could be broken after an exponential number of rounds [11]. Using a backward induction argument, they deduced that the participants would better keep silent at every round and thus the secret would never be reconstructed from the shares.

A different approach to tackle the problem of designing a rational SSS was taken by Lepinski *et al.* [12], Izmalkov *et al.* [13] and Micali and abhi shelat [14]. They

obtained rational MPC protocols secure against coalitions of adversaries. However, the hardware needed for these operations (secure envelopes and ballot boxes) is very restrictive and implementing such approaches is likely to be complicated.

## 1.2. Our results

As in the literature previously quoted, we design a protocol for rational threshold secret sharing. Our construction removes all of the drawbacks mentioned above. We neither assume an online dealer or any trusted parties (the mediator for example), nor do we rely on secure MPC to redistribute the shares of the secret. Instead, we borrow the idea from proactive SSS [15] to renew the shares by the interaction between players. Unlike constructions quoted above, the secret  $s$  is masked using a one-time pad. This provides information theoretical security and makes our construction immune to backward induction mentioned previously. Our scheme is based on symmetric polynomials. Even if this technique has already been used before for MPC protocols [16], to the best of our knowledge, it is the first time that they appear in rational cryptography. Our protocol is efficient in terms of interactions (as the dealer will not be needed), share size and computation and it guarantees that all players learn the secret at a Nash equilibrium whose strategy survives the iterated elimination of weakly dominated strategies. As in most of the prior work, we need a simultaneous broadcast channel and secure privacy channels. The protocol presented in this paper requires the threshold value  $m$  to be at least 4.

## 2. Game Theoretic Background

In this section, we present the game theoretic concepts our cryptographic construction relies on. As said in Sect.1.2, we assume the existence of simultaneous channels for each participant as well as the presence of private channels between any pair of players. We are to design a rational SSS with the expectation that, when rationally played, it opens the secret to all the players.

**Definition 1 ([17])** *An  $n$ -player game  $\Gamma = (\{A_1, \dots, A_n\}, \{U_1, \dots, U_n\})$ , presented in normal/standard form, is determined by specifying, for each player  $P_i$ , a set of possible actions  $A_i$  and a utility function  $U_i : A_1 \times \dots \times A_n \rightarrow \mathbb{R}$ . Any tuple of actions  $a := (a_1, \dots, a_n) \in A_1 \times \dots \times A_n$  is called an outcome.*

The utility function of each player expresses his preferred choices over outcomes.

**Definition 2 ([17])** *A player  $P_i$  prefers (resp. weakly prefers) outcome  $a$  to  $a'$  if and only if:  $U_i(a) > U_i(a')$  (resp.  $U_i(a) \geq U_i(a')$ ).*

The game  $\Gamma$  is played by having each party  $P_i$  choose an action  $a_i \in A_i$  and having all parties play their actions simultaneously. The *payoff* to  $P_i$  is the value given by his utility function:  $U_i(a_1, \dots, a_n)$ . The goal of each participant to the game is to maximize his utility function.

In order to obtain stable strategies (i.e. equilibria), some randomization in the choice of strategies is needed.

- Each player  $P_i$  chooses his action  $a_i \in A_i$  using a *distribution*  $\sigma_i$ .
- We are interested in *expected utilities* for each player.

**Definition 3 ([17])** Let  $\Gamma = (\{A_1, \dots, A_n\}, \{U_1, \dots, U_n\})$  be a game in normal form. Consider a tuple of strategy vectors  $\sigma = (\sigma_1, \dots, \sigma_n)$ .  $\sigma_i$  is a best response of  $P_i$  to  $\sigma_{-i}$  if it maximizes  $U_i(\sigma_i, \sigma_{-i})$  where  $\sigma_{-i}$  represents the  $(n - 1)$ -tuple of strategies played by the remaining players.

**Definition 4 ([17])** Let  $\Gamma = (\{A_1, \dots, A_n\}, \{U_1, \dots, U_n\})$  be a game in normal form and let  $\sigma_i$  be a distribution over  $A_i$ . A tuple  $\sigma = (\sigma_1, \dots, \sigma_n)$  is a mixed-strategy Nash equilibrium if for all  $i$  and every distribution  $\sigma'_i$  over  $A_i$ , we have:  $U_i(\sigma'_i, \sigma_{-i}) \leq U_i(\sigma)$ .

Intuitively, Definition 4 means that  $P_i$  has no incentive to deviate from  $\sigma_i$  as long as the remaining participants follow  $\sigma_{-i}$  (for all  $i \in \{1, \dots, n\}$ ). A Nash equilibrium formalizes a notion of rationality which is strictly internal: each player only cares about his own payoff.

**Theorem 5 (Nash [18])** Any game with a finite set of players and a finite set of strategies has a Nash equilibrium of mixed-strategies.

**Definition 6 ([19])** Given  $\Gamma = (\{A_1, \dots, A_n\}, \{U_1, \dots, U_n\})$ , we say that action  $a_i \in A_i$  is weakly dominated with respect to  $A_{-i} (= \prod_{j \neq i} A_j)$  if there exists a randomized strategy  $\sigma_i \in \Delta(A_i)$  such that:

- (1)  $\forall a_{-i} \in A_{-i} \quad u_i(\sigma_i, a_{-i}) \geq u_i(a_i, a_{-i})$ ,
- (2)  $\exists a_{-i} \in A_{-i} : u_i(\sigma_i, a_{-i}) > u_i(a_i, a_{-i})$ .

where  $\Delta(A_i)$  represents the set of strategies of player  $P_i$ .

The notion of Nash equilibrium is fundamental in game theory. In any Nash equilibrium, no player assigns positive probability to any strictly dominated action. Thus, any Nash equilibrium involving such a strategy will not occur in practice. As a consequence, for our cryptographic setting, we can purge those strategies out.

**Definition 7 ([19])** Given  $\Gamma = (\{A_1, \dots, A_n\}, \{U_1, \dots, U_n\})$  and  $\hat{A} \subset A$ , let  $\text{DOM}_i(\hat{A})$  denote the set of strategies in  $\hat{A}_i$  that are weakly dominated with respect to  $\hat{A}_{-i}$ . Set:

$$A_i^\infty := \bigcap_{k \geq 1} A_i^k \quad \text{where} \quad \forall k \geq 1 \quad A_i^k := A_i^{k-1} \setminus \text{DOM}_i(A_i^{k-1}).$$

A Nash equilibrium  $\sigma = (\sigma_1, \dots, \sigma_n)$  of  $\Gamma$  survives iterated deletion of weakly dominated strategies if  $\sigma_i \in \Delta(A_i^\infty)$  for all  $i \in \{1, \dots, n\}$ .

### 3. Our Protocol for $m$ -out-of- $n$ Rational Secret Sharing Secure against a Single Deviation

In order for the reader to get an easier understanding of our protocol, we first give a general view of our secret reconstruction phase in Sect. 3.1. The full description of our scheme is in Sect. 3.2.

#### 3.1. Overview of the reconstruction phase

Our scheme relies on the masking of the secret  $s$ . Such an approach already appeared in [10]. However, Fuchsbauer *et al.* used a verifiable random function (VRF) which is a cryptographic primitive, the existence of which is based on some computational assumption.

In order to provide information theoretical security, the dealer will first use a one-time pad  $r$  over the secret  $s$ . He will also mask  $r$  in a similar way with another random element  $r'$  and he will publish  $r + r'$  to a register accessible to all players. In order to recover  $s$ , the players will need to obtain both  $r$  and  $r'$ . That is why the second task of the dealer is to distribute  $r$  and  $r'$  amongst the  $n$  players using two independent instances of Shamir's scheme [1] with threshold  $m$ . Note that the public value  $r + r'$  will be used by the participants to check the consistency of the two reconstructions. The third task of the dealer consists of sharing  $s + r$  using a bivariate polynomial having degree  $m - 2$  in each of its two unknowns.

Assume that  $m^* (\geq m)$  players want to participate in the secret reconstruction process. We first consider the case where  $m$  is even. A similar construction holds when  $m$  is odd (see Sect. 3.5). Note that this differentiation "  $m$  is even/odd " has no influence on the dealer's job when initially sharing  $s$ .

The reconstruction phase proceeds in three stages. During the first two stages, the goal of the  $m^*$  players is to recover the pad  $r$  (using  $r'$  and  $r + r'$ ). The third stage is a sequence of "invalid" and "valid" iterations which is a frequently used technique for rational SSS. During each of these iterations, the broadcast shares correspond to  $s + r$ . Those iterations have the following properties:

- "invalid" iteration: no information about  $s$  is revealed since the number of shares related to  $s + r$  hold by each participant is less than the threshold value  $m - 1$ . At the end of such an iteration, shares are renewed.
- "valid" iteration: every player recovers  $s$  on the assumption that every participant follows the protocol (which will be demonstrated to be the case since they are rational).

The key in this process is the fact that nobody knows in advance whether the next iteration will be "valid".

During any iteration of the third stage, each of the  $m^*$  participating players  $P_{i_j}$  chooses a bit  $b_{i_j}$  such that  $b_{i_j} = 1$  with probability  $\alpha$  depending on the utilities of the  $n$  participants. Then, all  $m^*$  players commonly run a simple MPC protocol to compute the parity value  $p := b_{i_1} \oplus b_{i_2} \oplus \dots \oplus b_{i_{m^*}}$ . Our MPC protocol is an extension of what was done in [6] in the case of three players.

If  $p = 0$  then the  $m^*$  players are asked to repeat the previous iteration. Otherwise, each  $P_{i_j}$  broadcasts his share to the  $m^* - 1$  other members if  $b_{i_j} = 1$ .

In the case the protocol did not abort before this point, we have two possibilities:

- (1)  $P_{i_j}$  has at most  $m - 2$  shares (for some  $j$ ): the players run a check phase to catch potential cheaters. If the shares are correct, then the  $m^*$  players renew their shares of  $s$  using a technique from proactive SSS [15] and they start over by choosing a new random bit.
- (2) All players have at least  $m - 1$  shares: the set of  $m^*$  players attempt to reconstruct  $s + r$  using polynomial interpolation or error correcting techniques (see Sect. 3.2 for details). Once they obtain  $s + r$ , they can deduce  $s$  since they got  $r$  by the end of the second stage.

### 3.2. Construction

Our computations will be done in the finite field  $\mathbb{F}_q$  for which  $\omega$  is a primitive element. As mentioned earlier, we denote  $\mathcal{P} := \{P_1, \dots, P_n\}$  the set of participants and the secret value to be distributed is  $s \in \mathbb{F}_q$ . As said in the previous section, we consider that the threshold value  $m$  is even.

During the secret reconstruction phase, we assume the existence of a simultaneous broadcast channel for all participating players and the presence of private channels between any pair of these players. All these channels are authenticated.

For each  $i \in \{1, \dots, n\}$ , denote  $u_i$  (respectively,  $u_i^+$ ) the minimal (respectively, maximal) payoff of  $P_i$  when he retrieves the secret and denote  $u_i^-$  his maximal payoff when  $P_i$  does not recover  $s$ . As usually assumed in the rational cryptographic context, we consider:  $u_i^+ > u_i > u_i^-$  for all  $i \in \{1, \dots, n\}$ .

Without loss of generality, we also assume that for each player  $P_i$  ( $i \in \{1, \dots, n\}$ ), we have:

$$\frac{1}{q} u_i^+ + \left(1 - \frac{1}{q}\right) u_i^- < u_i \tag{1}$$

Assuming that Inequality (1) holds for all  $n$  players is *not* proper to our scheme. Indeed, this relation corresponds to the fact that it is more valuable for  $P_i$  to participate in the secret reconstruction process than aborting the protocol and tossing a coin to decide the value of the secret  $s$  since this selfish strategy is successful with probability  $\frac{1}{q}$ . As said in [10] about Fuchsbauer *et al.*'s value  $U_{\text{random}}$ , if Inequality (1) does not hold, then  $P_i$  has no incentive in cooperating at all and this player is better out of the group of participants.

#### 3.2.1. Initial share phase

This is the only phase where the dealer is active. His goal is to distribute  $s$  over  $\mathcal{P}$ .

- (1) The dealer chooses a two independent random values  $r$  and  $r'$  uniformly distributed over  $\mathbb{F}_q$ . The dealer publishes the value  $r + r'$  in a public register.

- (2) The dealer shares  $r$  into  $(r_1, \dots, r_n)$  and  $r'$  into  $(r'_1, \dots, r'_n)$  using two independent instances of Shamir's  $m$ -out-of- $n$  SSS. He distributes through a secure channel the pair  $(r_i, r'_i)$  to  $P_i$  for all  $i \in \{1, \dots, n\}$ .
- (3) Denote  $v := s + r$ . The dealer constructs symmetric bivariate polynomial  $f(x, y) = \sum_{i=0}^{m-2} \sum_{j=0}^{m-2} a_{ij} x^i y^j$  where  $a_{00} = v$ . For each  $i$ , the dealer sends the univariate polynomial  $h_i(x) := f(x, \omega^i)$  to  $P_i$  through a secure channel.

**Remark 8.** *Due to the symmetry of  $f$ , we have  $h_j(\omega^i) = h_i(\omega^j)$  for any pair  $(i, j)$ . This property is fundamental in our work.*

**Remark 9.** *During the initial share phase, the dealer uses a symmetric polynomial  $f$  to distribute the shares of  $v$ . We would like to emphasize why the degree of  $f$  in each of these two variable is  $m - 2$ . Since those shares are not authenticated by any cryptographic primitive, in order to verify the consistency of the data sent by a given  $P_i$  we require all remaining  $m^* - 1 (\geq m - 1)$  participants to run the check phase.*

### 3.2.2. Check phase

The goal of this phase is to check the consistency of the share  $\lambda$  broadcast by  $P_i$ . This task is done by the players participating in the secret reconstruction process – except  $P_i$ .

- (1) Each participating player  $P_j (j \neq i)$  broadcasts his check value  $h_j(\omega^i)$ .
- (2) Each of these players checks, using polynomial interpolation, whether the secret  $\lambda$  broadcast by  $P_i$  in Stage 3 of the secret reconstruction phase is consistent with those  $h_j(\omega^i)$ 's (i.e. they check if  $\lambda = h_i(0)$ ).

### 3.2.3. Share renewal phase

As the check phase, share renewal is done by the players participating in the secret reconstruction process. We assume that there are  $m^* (\geq m)$  such players. For ease of description, we can assume without loss of generality that those players are  $P_1, P_2, \dots, P_{m^*}$ .

In this phase, each participating  $P_i$  plays a similar role to the dealer's (initial share phase) to renew his share for  $v$ .

- (1) Each  $P_i$  selects a random symmetric polynomial  $\delta_i(x, y)$  of degree  $m - 3$ . He sends  $\delta_{i,j}(x) := \delta_i(x, \omega^j)$  to  $P_j$  over a private channel (for all  $j \in \{1, \dots, m^*\} \setminus \{i\}$ ).
- (2) After receiving  $\delta_{i,j}(x)$ ,  $P_j$  computes and sends the value  $\delta_{i,j}(\omega^k)$  to  $P_k$ , for  $k \in \{1, \dots, m^*\} \setminus \{j\}$ , over a private channel.
- (3)  $P_k$  checks whether  $\delta_{i,j}(\omega^k) = \delta_{i,k}(\omega^j)$ , for  $j = 1, \dots, m^*$ .
- (4) If one of these equalities is not satisfied for some pair  $(i, j)$ , then  $P_k$  aborts the whole protocol. Otherwise, each  $P_j$  updates his share as:  $h_j(x) \leftarrow h_j(x) + (x +$

$$\omega^j \sum_{i=1}^{m^*} \delta_{i,j}(x).$$

**Remark 10.** *After the renewal phase, we have the following relation for the new shares:  $h_j(\omega^k) = h_k(\omega^j)$  for any  $1 \leq j, k \leq m^*$ .*

### 3.2.4. Secret reconstruction phase

We assume that  $m^*(\geq m)$  players participate in the secret reconstruction. As before, we can assume that they are  $P_1, P_2, \dots, P_{m^*}$ . Our reconstruction protocol contains three stages for each of these  $m^*$  players. The first stage is dedicated to the recovery of the random value  $r$  used by the dealer as a pad over the secret  $s$ .

#### Stage 1

- (1) Each  $P_i$  broadcasts his pair  $(r_i, r'_i)$ . If some player  $P_j$  obtains less than  $m^*$  pairs (including his own), then  $P_j$  aborts the whole protocol.
- (2) Each  $P_i$  constructs two sets of shares. The first one  $S_{i,r}$  consists of all the first components of those  $m^*$  pairs (i.e. the  $r_i$ 's) and the second set  $S_{i,r'}$  contains all the second components of the pairs (i.e. the  $r'_i$ 's). Player  $P_i$  checks if each of these sets can be interpolated by a polynomial of degree at most  $m - 1$ . If this checking process is unsuccessful for some  $P_j$ 's, then  $P_j$  stops the protocol.
- (3) For each  $i \in \{1, \dots, n\}$ , we denote  $\mathcal{R}_i$  (respectively  $\mathcal{R}'_i$ ) the constant term of the polynomial reconstructed by  $P_i$  corresponding to the set  $S_{i,r}$  (respectively  $S_{i,r'}$ ). Each  $P_i$  checks whether the sum  $\mathcal{R}_i + \mathcal{R}'_i$  is equal to the public value  $r + r'$ .
  - If the verification is unsuccessful for some  $P_j$ , then he aborts the protocol.
  - Otherwise, all participants proceed to Stage 2.

The remaining two stages are used to recover  $v$ . Note that the threshold now is  $m - 1$  rather than  $m$  since the symmetric bivariate polynomial  $f$  has degree  $m - 2$  in each of its variables.

#### Stage 2

- (1) Each  $P_i$  chooses a bit  $b_i$  with  $\Pr(b_i = 1) = \alpha$  as well as a uniformly distributed random bit  $b'_i$ .
- (2) Denote  $d_i := b_i \oplus b'_i$ . Let  $i^+$  denote  $i + 1$  except that  $(m^*)^+$  is 1. Similarly  $i^-$  denotes  $i - 1$  except that  $1^-$  is  $m^*$ . Each  $P_i$  sends  $b'_i$  to player  $P_{i^+}$  and  $d_i$  to player  $P_{i^-}$  using private channels. If some  $P_j$  does not send data to both neighbors  $P_{j^-}$  and  $P_{j^+}$ , then the protocol aborts.
- (3) Each  $P_i$  computes and broadcasts  $b'_{i^-} \oplus d_{i^+}$ . If some  $P_j$  does not receive the bits as prescribed, then the protocol aborts. Otherwise, denote  $\Delta_{1,i}, \dots, \Delta_{m^*,i}$  the  $m^*$  elements collected by  $P_i$  during this broadcast including his own. Each  $P_i$  computes  $p_i = \bigoplus_{j=1}^{m^*} \Delta_{j,i}$ .



- (4) If  $p_j = 0$  for some  $P_j$ , then the whole protocol goes back to the first step of Stage 2. Otherwise, all participants proceed to Stage 3.

**Remark 11.** *When the  $m^*$  players reach Stage 3, then we have:  $\forall i \in \{1, \dots, m^*\}, p_i = 1$ . Requiring that all the  $p_i$ 's be equal to 1 means that each participant is holding an odd number of shares since  $m$  is even. This is due to the fact that  $m - 1$  (odd number) is the minimum number of shares that a participant needs to uniquely determine a polynomial of degree  $m - 2$ .*

**Stage 3**

- (1) Each  $P_i$  broadcasts his share  $h_i(0)$  if  $b_i = 1$ . Denote  $k_i$  the number of shares that  $P_i$  received during the previous broadcast (including his own if  $b_i = 1$ ).
- (2) If  $k_i < m - 1$  for at least  $m^* - 1$  players  $P_i$ , then:
  - (a) If  $k_i$  is even for at least  $m^* - 1$  players  $P_i$ , then the protocol stops.
  - (b) Otherwise, all  $m^*$  active players participate in the check phase. If some  $P_j$  does not broadcast  $h_j(\omega^j)$  as required, then the protocol aborts. Otherwise, all players go to the renewal phase and then they proceed to the beginning of Stage 2.
- (3) If  $k_i \in \{m - 1, m\}$  for at least  $m^* - 1$  players  $P_i$ , then each of these players  $P_i$  interpolates the shares into a polynomial  $f_i(0, y)$ . If the degree of  $f_i(0, y)$  is  $m - 1$  then  $P_i$  aborts the protocol. Otherwise, he outputs  $f_i(0, y) + \mathcal{R}_i$  where  $\mathcal{R}_i$  was computed at the third step of Stage 1. After this computation, the protocol ends.
- (4) If  $k_i \geq m + 1$  for at least  $m^* - 1$  players  $P_i$ , then each of these players  $P_i$  chooses any  $(m + 1)$ -subset  $h_{i,1}, \dots, h_{i,m+1}$  of his  $k_i$  values. Each  $P_i$  forms a  $(m + 1)$ -vector  $(h_{i,1} \cdots h_{i,m+1})$  which is decoded using a  $[m + 1, m - 2, 3]$  generalized Reed-Solomon (GRS) decoder. Finally, each  $P_i$  extracts the secret value  $s$  using the previous corrected codeword and  $\mathcal{R}_i$  and the protocol ends.

**Remark 12.** *Our choice of  $k_i$  at step 1 of Stage 3 is to insure that all participants following the protocol's instructions will obtain the same value  $k_i = k$ . This value  $k$  represents the number of elements which were broadcast.*

**Remark 13.** *The goal of the check phase played at step 2 of Stage 3 is to punish a single deviating player since, in such a case, the protocol would abort and no one would learn  $s$ . Since we use a simultaneous broadcast channel to send data (step 1 of Stage 3), no players know whether the next iteration will correspond to step 2 or step 3/4 - called "invalid-valid" in Sect. 3.1 - before data transmission. Furthermore, step 2.b does not reveal anything about  $v$  since the check phase run to check the consistency of at most  $m - 2$  values.*

**Remark 14.** *Since that  $m \neq q - 1$  (in fact, we have  $q \geq n$ ), we cannot use Reed-Solomon codes [20] but we have to work with their generalized form [21].*

**Remark 15.** *The use of a GRS decoder at step 4 of Stage 3 is due to the fact that  $(h_1(0), \dots, h_n(0))$  can be interpreted as a sharing of  $s + r$  using Shamir’s technique with threshold parameter  $m - 1$  and the well-known relation between Shamir’s SSS and GRS codes [20]. Note that we cannot use the Lagrange interpolation technique since we need to ensure correct secret reconstruction in the presence of (at most) one deviating player (see proof of Theorem 17 (Stage 3)).*

**Remark 16.** *Our protocol requires  $m \geq 3$ . Indeed, consider  $m^* = m = 2$  and two participating players  $P_1$  and  $P_2$ . If  $P_1$  always remains silent at step 1 of Stage 3 (even if  $b_1 = 1$ ) then  $P_2$  is forced to permanently run step 2.b. Since  $P_2$  follows the protocol faithfully, at some iteration, his share is to be broadcast to  $P_1$  (who will still be silent). Thus,  $P_1$  will recover  $s$  and  $P_2$  will not. To prevent a player to choose such a strategy, we need  $m - 1 > 1$ . That is  $m \geq 3$ .*

### 3.3. Security of our rational SSS

We introduce the following notations:

- The secret reconstruction protocol is denoted  $\Pi(\alpha)$ .
- As said at the beginning of Sect. 3.2, for each  $i \in \{1, \dots, n\}$ , denote  $u_i$  (respectively,  $u_i^+$ ) the minimal (respectively, maximal) payoff of  $P_i$  when he retrieves the secret and denote  $u_i^-$  his maximal payoff when  $P_i$  does not recover  $s$ . As usually assumed in the rational context, we consider:  $u_i^+ > u_i > u_i^-$  for all  $i \in \{1, \dots, n\}$ .

The following theorem shows the consistency of our scheme in a rational environment. Its proof can be found in Appendix A.

**Theorem 17.** *Assume  $m \geq 4$ . There exists an  $\alpha^*$  such that for any  $\alpha \leq \alpha^*$ ,  $\Pi(\alpha)$  induces a Nash equilibrium surviving iterated deletion of weakly dominated strategies.*

**Remark 18.** *We do not say that our prescribed strategy does not lead to a Nash equilibrium surviving iterated deletion of weakly dominated strategies when  $m = 3$  (see Remark 16 for the case  $m = 2$ ). Our demonstration simply does not handle this case for which our work is still open.*

Explicitly computing the largest possible value for  $\alpha^*$  is not trivial. Fortunately, we can obtain an explicit bound more easily. We define the values  $\beta := \max_{i=1, \dots, n} \frac{u_i^+ - u_i}{u_i - u_i^-}$

and  $\tilde{\alpha} := \left[ \left( \frac{m^*}{m-3} - 1 \right) \sqrt{\beta + 1} \right]^{-1}$ . The proof of the following result can be found in Appendix B.

**Theorem 19.** *Assume  $m \geq 4$ . For any  $\alpha \leq \tilde{\alpha}$ ,  $\Pi(\alpha)$  induces a Nash equilibrium surviving iterated deletion of weakly dominated strategies.*

**3.4. Round complexity**

We use the same notations as in the previous section.

**Upper Bound.** The following result is based on the fact:

$$\Pr(s \text{ is recovered}) \geq \sum_{\substack{\lambda = m-1 \\ \lambda \text{ odd}}}^{m^*} \alpha^{m^*} \binom{m^*}{\lambda} \geq m^* \alpha^{m^*}$$

This is easy to demonstrate and, due to space limitations, we skipped those details.

**Theorem 20.** Assume  $m \geq 4$ . Let  $\alpha^*$  be as in Theorem 17. For any  $\alpha \leq \alpha^*$ , the expected round complexity of  $\Pi(\alpha)$  is:

$$\frac{1}{\sum_{j=\frac{m}{2}}^{\lceil \frac{m^*}{2} \rceil} \alpha^{2j-1} (1-\alpha)^{m^*-(2j-1)} \binom{m^*}{2j-1}}$$

which is  $O(\frac{\alpha^{-m^*}}{m^*})$ .

**Lower Bound.** We now show a lower bound on the round complexity of our protocol. The demonstration of the following theorem relies on the fact:

$$\begin{aligned} \Pr(s \text{ is recovered}) &= (1-\alpha) \sum_{\substack{\lambda = m-2 \\ \lambda \text{ odd}}}^{m^*-1} \alpha^\lambda (1-\alpha)^{m^*-1-\lambda} \binom{m^*-1}{\lambda} \\ &+ \\ &\alpha \sum_{\substack{\lambda = m-2 \\ \lambda \text{ even}}}^{m^*-1} \alpha^\lambda (1-\alpha)^{m^*-1-\lambda} \binom{m^*-1}{\lambda} \end{aligned}$$

and on the Chernoff’s bound on the tail of the binomial distribution [22]. Since space is limited, we did not include the details here.

**Theorem 21.** Assume  $m \geq 4$ . For any  $\alpha \leq \min(\alpha^*, \frac{m-2}{m^*-1})$ , the expected round complexity of  $\Pi(\alpha)$  is:

$$\Omega \left( \left( \frac{m-2}{m^*-1} \alpha^{-1} \right)^{m-2} \frac{e^{\alpha(m^*-1)-m+2}}{1-\alpha} \right)$$

**3.5. Remark on the case  $m$  is odd**

Since the beginning of Sect. 3, we only considered the case when  $m$  was even. When the threshold  $m$  is odd, we can essentially use the same protocol with the exception of step 4 in Stage 2 and step 2.a in Stage 3 which become:

**Stage 2 (update)**

4. If  $p_j = 1$  for some  $P_j$ , then the whole protocol goes back to the first step of Stage 2. Otherwise, all participants proceed to Stage 3.

**Stage 3 (update)**

- 2.a. If  $k_i$  is odd for at least  $m^* - 1$  players  $P_i$ , then the protocol stops.

It holds similar security and efficiency theorems to those presented in the past two sections. The only analytical difference lies in the fact that we now have  $p = 0$ .

**3.6. Discussion**

**Equilibrium.** Our solution concept is based on equilibria surviving iterated deletions of weakly dominated strategies. The study of this type of equilibrium was introduced by Halpern and Teague [6] and has received a lot of attention [7, 9, 23]. We are aware that the notion of iterated deletion exhibits several problems [24] and that several new concepts have been proposed (mainly using computational versions of Nash equilibria [10, 11]). The purpose of this paper is not to advocate in favor of a specific type of equilibrium. Its primary goal is to present a new construction combining the advantages of several schemes for a model widely studied in the literature.

**Communication Channels.** Our rational protocol requires the presence of simultaneous broadcast channels which is a commonly-used model for rational SSS. In [11], Kol and Naor managed to remove the need of simultaneity for the broadcast channels. However, this is at the expense of increasing the round complexity by a multiplicative  $m$  and the removal is based on permutations to relocate the meaningful encryption key. Thus, this process is related to use of their meaningful/meaningless encryption primitive. In [10], Fuchsbauer *et al.* only used point-to-point channels. However, authentication needs to use the VRF.

**Share Consistency.** We use an error-correcting code to check the consistency of the shares. Another possibility would have been to use information theoretically secure MACs as in [24]. The issue is that these MACs encounter a small authentication error probability while our coding-based approach does not.

**Computation Efficiency.** Several rational protocols (such as [6]) require the dealer to participate in every round of the secret reconstruction phase. This is a bottleneck for the efficiency of those constructions. Like [10, 11], our scheme does not require the presence of an online dealer. Furthermore, our share renewal process does not rely on either complex MPC protocols (contrary to [7, 8, 9]) or complicated hardware such as envelopes and ballot boxes (contrary to [13, 14]). In terms of round complexity, our protocol exhibits an expected value  $O(\frac{\alpha^{-m^*}}{m^*})$  where the number was  $5\alpha^{-3}$  for 3-out-of-3 secret sharing [6]. In addition, in [6], the protocol security is computational (while it is information theoretic for our protocol) and the dealer interacts with the payers at every round.

**Backward Induction.** In [11], Kol and Naor emphasized that techniques from [6, 9, 7] were susceptible to backward induction attacks resulting in all players remaining silent from the beginning of the secret reconstruction process. This attack requires an exponential number of rounds to succeed. Such large running time only occurs with negligible probability. Nonetheless, our scheme is immune against this threat since we only use information theoretical tools (one-time pads) to authenticate data. In particular, our immunity does not require the existence of any additional cryptographic primitive contrary to [11] where meaningful/meaningless encryption schemes were used.

#### 4. Conclusion

In this paper, we presented a new protocol for rational threshold secret sharing based on symmetric polynomials. To the best of our knowledge, it is the first time that such polynomials have been used for rational secret sharing. Our protocol requires simultaneous broadcast channels and  $m \geq 4$ . This construction does not require the presence of the dealer during the share reconstruction phase and it provides information theoretical security. It is immune against the backward induction attack and it leads to a Nash equilibrium surviving the iterated deletion of weakly dominated strategies.

On the negative side, our scheme is only secure against single strategy deviations. One line to follow for our future research is to extend this scheme to handle the case of coalition of enemies ( $c$ -resilience for  $c \geq 2$ ). Using bivariate polynomials of degree  $m - 2c$  is a possible approach as they would also use the GRS code to correct up to  $c$  errors. It is not hard to see that the information theoretical security provided by the one-time pads still holds (for any coalition of size at most  $\frac{m-1}{2}$ ). Furthermore, the check phase would still be consistent as every player  $P_j$  (testing the validity of  $\lambda$  sent by  $P_i$ ) would get  $m^* - c \geq m - 2c + 1$  correct broadcast elements at the end of step 1. The tricky point with this approach is to perform the probabilistic analysis of a group of  $c$  cheaters. Indeed, the probabilistic formulas exposed in Appendix A cannot be simplified as easily since we have to handle a set of  $c$  bits which may not have been chosen by the cheaters as stated in the algorithm. In the case  $c = 1$ , we obtained simple conditional probabilities since we represented the bit of the players following the protocol's instructions. This is no longer the case for coalitions of size  $c \geq 2$  as some cheaters may choose their bit as they feel best for themselves.

As said in Sect. 3.6, our solution concept is based on equilibria surviving iterated deletions of weakly dominated strategies and several other approaches in designing rational protocols have recently been proposed. Since the cryptographic community is still in search of a proper framework for rational protocols, it would be interesting to study the benefits of our approach in different equilibrium contexts.

#### Acknowledgment

The authors would like to thank the reviewers for their suggestions to improve the quality of this paper. Christophe Tartary's work was funded by the National Science

Foundation of China grants 61033001, 61061130540, 61073174 and 61050110147 (International Young Scientists program) as well as the National Basic Research Program of China grants 2007CB807900 and 2007CB807901. Christophe Tartary also acknowledges support from the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the theory of Interactive Computation (CTIC) within which part of this work was performed. The work of Huaxiong Wang was supported by the Singapore National Research Foundation under Research grant NRF-CRP2-2007-03.

## References

- [1] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612 – 613, November 1979.
- [2] G. R. Blakley. Safeguarding cryptographic keys. In *AFIPS 1979 National Computer Conference*, pages 313 – 317. AFIPS Press, 1979.
- [3] D. Catalano, R. Cramer, I. Damgård, G. Di Crescenzo, D. Poincheval, and T. Takagi. *Contemporary Cryptology*. Advanced Courses in Mathematics - CRM Barcelona. Birkhäuser, 2005.
- [4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th Annual ACM Symposium on Theory of Computing*, pages 1 – 10. ACM Press, 1988.
- [5] C. Crépeau, D. Gottesmany, and A. Smith. Secure multiparty quantum computation. In *34th Annual ACM symposium on Theory of Computing*, pages 643 – 652. ACM Press, 2002.
- [6] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: Extended abstract. In *36th Annual ACM Symposium on Theory of Computing*, pages 623 – 632. ACM Press, 2004.
- [7] 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 Annual ACM Symposium on Principles of Distributed Computing*, pages 53 – 62. ACM Press, 2006.
- [8] G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. In *Crypto'09*, volume 5677 of *LNCS*, pages 559 – 576. Springer - Verlag, 2009.
- [9] S. Dov Gordon and J. Katz. Rational secret sharing, revisited. In *5th International Conference on Security and Cryptography for Networks*, volume 4116 of *LNCS*, pages 229 – 241. Springer - Verlag, 2006.
- [10] G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In *7th Theory of Cryptography Conference*, volume 5978 of *LNCS*, pages 419 – 436. Springer - Verlag, 2010.
- [11] G. Kol and M. Naor. Cryptography and game theory: Designing protocols for exchanging information. In *5th Theory of Cryptography Conference*, volume 4948 of *LNCS*, pages 320 – 339. Springer - Verlag, 2008.
- [12] M. Lepinski, S. Micali, and abhi shelat. Collusion-free protocols. In *37rd Annual ACM Symposium on Theory of Computing*, pages 543 – 552. ACM Press, 2005.
- [13] S. Izmalkov, S. Micali, and M. Lepinski. Rational secure computation and ideal mechanism design. In *46th Annual Symposium on the Foundations of Computer Science*, pages 585 – 594. IEEE Computer Society, 2005.

- [14] S. Micali and abhi shelat. Purely rational secret sharing (extended abstract). In *6th Theory of Cryptography Conference*, volume 5444 of *LNCS*, pages 54 – 71. Springer - Verlag, 2009.
- [15] P. D’Arco and D. R. Stinson. On unconditionally secure robust distributed key distribution centers. In *Asiacrypt’02*, volume 2501 of *LNCS*, pages 346 – 363. Springer - Verlag, 2002.
- [16] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *Eurocrypt’99*, volume 1592 of *LNCS*, pages 311 – 326. Springer - Verlag, 1999.
- [17] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [18] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286 – 295, 1951.
- [19] J. Katz. Bridging game theory and cryptography: Recent results and future directions. In *5th Theory of Cryptography Conference*, volume 4948 of *LNCS*, pages 251 – 272. Springer - Verlag, 2008.
- [20] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of Society for Industrial and Applied Mathematics*, 8(2):300 – 304, June 1960.
- [21] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*, volume 16 of *North-Holland Mathematical Library*. North-Holland, 1977.
- [22] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493 – 507, December 1952.
- [23] A. Lysyanskaya and N. Triandopoulos. Rationality and adversarial behavior in multiparty computation. In *Crypto’06*, volume 4117 of *LNCS*, pages 180 – 197. Springer - Verlag, 2006.
- [24] G. Kol and M. Naor. Games for exchanging information. In *40th Annual ACM Symposium on Theory of Computing*, pages 423 – 432. ACM Press, 2008.

## Appendix A. Proof of Theorem 17

We first show that the recommended protocol is a Nash equilibrium for  $\Pi(\alpha)$ .

Without loss of generality, we can assume that the active players are  $P_1, \dots, P_{m^*}$ . If one of the  $m^*$  players does not broadcast anything during step 1 of Stage 1, then the protocol would terminate without nobody recovering  $s$ . This would result in a lower payoff for everybody including the deviating player. Thus, any rational player is to broadcast a value during that step. Suppose that some  $P_i$  broadcasts a fake couple  $(\hat{r}_i, \hat{r}'_i)$  while the others follow the prescribed strategy. In order for the protocol not to abort, this forgery must lead to a couple of values  $(\widehat{\mathcal{R}}, \widehat{\mathcal{R}}')$  such that:  $\widehat{\mathcal{R}} + \widehat{\mathcal{R}}' = r + r'$  (random element uniformly distributed over  $\mathbb{F}_q$ ). In other words, the deviation of  $P_i$  is successful if  $(\hat{r}_i, \hat{r}'_i)$  corresponds to one of the  $q - 1$  couples  $(\widehat{\mathcal{R}}, \widehat{\mathcal{R}}')$  consistent with  $r + r'$  and with  $\hat{r}_i \neq r_i$ .

**Remark 1.** *We stress that the values  $\widehat{\mathcal{R}}$  and  $\widehat{\mathcal{R}}'$  are common to all  $m^* - 1$  honest players.*

Since we use simultaneous broadcast channels, deviating  $P_i$  must input  $(\hat{r}_i, \hat{r}'_i)$  (with  $\hat{r}_i \neq r_i$ ) before receiving any information about the shares of the honest players. As a consequence, the result of the polynomial reconstructions appears uniformly

distributed to  $P_i$ . Thus, the value  $\widehat{\mathcal{R}} + \widehat{\mathcal{R}}'$  appears uniformly distributed over  $\mathbb{F}_q$ . Therefore, the expected payoff of  $P_i$  by performing this deviation is:

$$\frac{1}{q} u_i^+ + \left(1 - \frac{1}{q}\right) u_i^-$$

Thus,  $P_i$  does *not* deviate if:

$$\frac{1}{q} u_i^+ + \left(1 - \frac{1}{q}\right) u_i^- < u_i$$

which is Inequality (1) stated at the beginning of Sect. 3.2. As a consequence, any rational player is to follow all instructions in Stage 1.

Assume that player  $P_i$  wants to cheat during Stage 2. In order for the protocol not to be terminated abruptly,  $P_i$  is to send data to  $P_{i^+}$  and  $P_{i^-}$  at step 1 and to all other  $(m^* - 1)$  players at step 3. A necessary condition for a successful cheating is to have:

$$p_1 = p_2 = \dots = p_{i^-} = p_{i^+} = \dots = p_{m^*} = 1$$

Denote  $\widetilde{d}_i, \widetilde{b}'_i$  and  $\widetilde{d_{i^+} \oplus b'_{i^-}}$  the values sent by the deviating  $P_i$  during Stage 2.

Let  $j$  be any value in  $\{1, \dots, m^*\} \setminus \{i\}$ . We first study  $p_j$ . Following the protocol's instructions,  $P_j$  computes the bit  $p_j$  as:

$$\begin{aligned} p_j &= \underbrace{(b'_{i^-} \oplus \widetilde{d}_i)}_{\text{from } P_{i^-}} \oplus \underbrace{(d_{i^+} \oplus \widetilde{b}'_{i^-})}_{\text{from } P_i} \oplus \underbrace{(\widetilde{b}'_i \oplus d_{i^{++}})}_{\text{from } P_{i^+}} \oplus \underbrace{\left[ \bigoplus_{\substack{k=1 \\ k \neq i^-, i, i^+}}^{m^*} (b'_{k^-} \oplus d_{k^+}) \right]}_{\text{from remaining players including } P_j} \\ &= (b'_{i^+} \oplus d_{i^-}) \oplus (\widetilde{d}_i \oplus \widetilde{b}'_i) \oplus (d_{i^+} \oplus \widetilde{b}'_{i^-}) \oplus \left[ \bigoplus_{\substack{k=1 \\ k \neq i^-, i, i^+}}^{m^*} b_k \right] \end{aligned}$$

We notice that the value of  $p_j$  does not depend on the index  $j$ . Therefore, we rename this common value as  $p$ . It can be simplified as follows:

$$p = \underbrace{(d_{i^+} \oplus b'_{i^-})}_{\text{known to } P_i} \oplus \underbrace{(\widetilde{d}_i \oplus \widetilde{b}'_i)}_{\text{chosen by } P_i} \oplus \underbrace{(d_{i^+} \oplus \widetilde{b}'_{i^-})}_{\text{chosen by } P_i} \oplus \left[ \bigoplus_{\substack{k=1 \\ k \neq i}}^{m^*} b_k \right] \tag{A.1}$$

Denote  $w_i$  the Hamming weight of the vector  $(b_1, \dots, b_{i^-}, b_{i^+}, \dots, b_{m^*})$ . We have two cases to consider:

- (1)  $\widetilde{d}_i, \widetilde{b}'_i$  and  $\widetilde{d_{i^+} \oplus b'_{i^-}}$  are such that:  $p = 1 \oplus \left[ \bigoplus_{\substack{k=1 \\ k \neq i}}^{m^*} b_k \right]$ .
- (2)  $\widetilde{d}_i, \widetilde{b}'_i$  and  $\widetilde{d_{i^+} \oplus b'_{i^-}}$  are such that:  $p = \bigoplus_{\substack{k=1 \\ k \neq i}}^{m^*} b_k$ .

Case 1. Since  $p = 1$ ,  $w_i$  is even. If the number of honest participating players to send their shares at step 1 of Stage 3 is either at most  $m - 4$  or at least  $m$ , then  $P_i$  does



not gain anything by deviating.  $P_i$  only benefits from not following the protocol's instructions when there are exactly  $m - 2$  honest participating players. Then,  $P_i$ 's expected payoff is  $u_i^+$ .

**Case 2.** Since  $p = 1$ ,  $w_i$  is odd. If the number of honest participating players to send their shares at step 1 of Stage 3 is either at most  $m - 3$  or at least  $m + 1$ , then  $P_i$  does not gain anything by deviating.  $P_i$  only benefits from not following the protocol's instructions when there are exactly  $m - 1$  honest participating players. Then,  $P_i$ 's expected payoff is  $u_i^+$ .

Reaching this point in our reasoning, we need to reflect upon  $P_i$ 's strategy. We showed that he only has incentive to deviate in two specific subcases of Case 2 (assuming that  $p = 1$ ):  $w_i = m - 2$  and  $w_i = m - 1$ . However, if  $P_i$  follows the instructions at Stage 2, then each of the remaining  $(m^* - 1)$  players would get:

$$p = \bigoplus_{k=1}^{m^*} b_k \tag{A.2}$$

So,  $P_i$  would obtain the same benefit in cheating at step 1 of Stage 3 as in the subcases above where the role of  $(d_{i+} \oplus b'_{i-}) \oplus (\tilde{d}_i \oplus \tilde{b}'_i) \oplus (\widetilde{d_{i+} \oplus b'_{i-}})$  from Eq. (A.1) would be played by  $b_i$  in Eq. (A.2). Therefore,  $P_i$  has no incentive in sending different values than those prescribed by the protocol during Stage 2.

We now focus on potential deviations of  $P_i$  during Stage 3. There are three possibilities for  $P_i$  to cheat at step 1 of Stage 3 (where  $p = 1$ ):

- (1)  $P_i$  broadcasts some value despite  $b_i = 0$ .
- (2)  $P_i$  does not broadcast anything despite  $b_i = 1$ .
- (3)  $P_i$  broadcasts a fake share when  $b_i = 1$ .

**Case 1.** The protocol will terminate during this iteration of Stage 3 since the value  $k_j$  will be even for  $j \neq i$  (step 2.b cannot be accessed). In this situation,  $P_i$  is better sending a fake share value. Since  $p = 1$ ,  $w_i$  is odd. We are in the same situation as in Case 2 of Stage 2 where  $P_i$  sends a fake share.

As said above, if  $w_i \leq m - 3$  then no players learn  $s$  and  $P_i$ 's expected payoff is  $u_i^-$ .

If  $w_i \geq m + 1$  then everybody recovers  $s$ :

- $P_i$  interpolates  $m - 2$  of the  $w_i$  shares he got from the other players (he runs step 3).
- Each  $P_j$  ( $j \neq i$ ) is to execute step 4. Since there is at most one incorrect value amongst the  $(m + 1)$  elements he chooses, this potential error is to be corrected by the GRS decoder and  $P_j$  recovers  $s$ .

In this situation, the expected payoff of the cheating  $P_i$  is  $u_i$ .

If  $w_i = m - 1$  then only  $P_i$  will recover  $s$  as the remaining players will reconstruct an incorrect polynomial at step 3. With large probability, its constant term will be different from  $s + r$ . In such a situation,  $P_i$  gets at most  $u_i^+$ . We need to compute

the following three probabilities:

$$\begin{aligned} & \Pr(w_i \geq m + 1 | \{p = 1\} \cap \{b_i = 0\}) \\ & \Pr(w_i = m - 1 | \{p = 1\} \cap \{b_i = 0\}) \\ & \Pr(w_i \leq m - 3 | \{p = 1\} \cap \{b_i = 0\}) \end{aligned}$$

Let  $\lambda$  be any element of  $\{0, \dots, m^* - 1\}$ .

$$\begin{aligned} \Pr(w_i = \lambda | \{p = 1\} \cap \{b_i = 0\}) &= \frac{\Pr(\{w_i = \lambda\} \cap \{\bigoplus_{k=1}^{m^*} b_k = 1\} \cap \{b_i = 0\})}{\Pr(\{\bigoplus_{k=1}^{m^*} b_k = 1\} \cap \{b_i = 0\})} \\ &= \frac{\Pr(\{w_i = \lambda\} \cap \{w_i \text{ is odd}\})}{\Pr(w_i \text{ is odd})} \\ &= \begin{cases} 0 & \text{if } \lambda \text{ is even} \\ \frac{\Pr(w_i = \lambda)}{\Pr(w_i \text{ is odd})} & \text{if } \lambda \text{ is odd} \end{cases} \end{aligned}$$

We can now compute the probabilistic values we need using the fact that  $m$  is even.

$$\begin{aligned} \Pr(w_i \geq m - 1 | \{p = 1\} \cap \{b_i = 0\}) &= \frac{\sum_{\substack{\lambda = m + 1 \\ \lambda \text{ odd}}}^{m^* - 1} \Pr(w_i = \lambda)}{\Pr(w_i \text{ is odd})} \\ \Pr(w_i \leq m - 3 | \{p = 1\} \cap \{b_i = 0\}) &= \frac{\sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m - 3} \Pr(w_i = \lambda)}{\Pr(w_i \text{ is odd})} \end{aligned}$$

Based on this analysis,  $P_i$  is **not** to cheat if:

$$u_i^+ \Pr(w_i = m - 1) + u_i \sum_{\substack{\lambda = m + 1 \\ \lambda \text{ odd}}}^{m^* - 1} \Pr(w_i = \lambda) + u_i^- \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m - 3} \Pr(w_i = \lambda) \leq u_i \Pr(w_i \text{ is odd})$$

The previous inequality is equivalent to:

$$\begin{aligned} & u_i^+ \alpha^{m-1} (1 - \alpha)^{m^* - m} \binom{m^* - 1}{m - 1} + u_i^- \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m - 3} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda + 1)} \binom{m^* - 1}{\lambda} \\ & \leq u_i \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m - 1} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda + 1)} \binom{m^* - 1}{\lambda} \end{aligned}$$

Since  $m$  is even, the sum on the right hand side ends when  $\lambda = m - 3$ . We get:

$$(u_i^+ - u_i)\alpha^{m-1}(1 - \alpha)^{m^* - m} \binom{m^* - 1}{m - 1} \leq (u_i - u_i^-) \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m-3} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda + 1)} \binom{m^* - 1}{\lambda}$$

We divide both sides of the previous inequality by  $\alpha^{m-1} (1 - \alpha)^{m^* - m} (u_i - u_i^-)$ . Defining  $\mathcal{A} := \frac{1 - \alpha}{\alpha}$ , we obtain:

$$\frac{u_i^+ - u_i}{u_i - u_i^-} \binom{m^* - 1}{m - 1} \leq \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m-3} \binom{m^* - 1}{\lambda} \mathcal{A}^{m - (\lambda + 1)} \tag{A.3}$$

The right hand side of Inequality(A.3) is a polynomial of degree  $m - 2$  in  $\mathcal{A}$  with a positive leading coefficient as soon as  $m - 3 \geq 1$  (i.e.  $m \geq 4$ ). Since  $\mathcal{A} \xrightarrow{\alpha \rightarrow 0_+} +\infty$ , we deduce that there exists a value  $\alpha_{i,1}$  such that for all  $\alpha \leq \alpha_{i,1}$ , Inequality(A.3) does hold. In such a situation,  $P_i$  does **not** cheat as indicated in Case 1.

Case 2. As in Case 1, the protocol will terminate during this iteration of Stage 3. In this case, both  $m$  and  $w_i$  are even. We are in the same situation as in Case 1 of Stage 2 where  $P_i$  remains silent.

If  $w_i \leq m - 4$  then no players learn  $s$  and  $P_i$ 's expected payoff is  $u_i^-$ . If  $w_i \geq m$  then everybody runs step 3 and recovers  $s$  since all the shares are genuine. The expected payoff of the cheating  $P_i$  is  $u_i$ . When  $w_i \leq m - 2$ ,  $P_i$  will be the only player to recover  $s$  since all other participants will run step 2.a. In this situation,  $P_i$  gets at most  $u_i^+$ . We are interested in the following three probabilities:

$$\begin{aligned} & \Pr(w_i \geq m | \{p = 1\} \cap \{b_i = 1\}) \\ & \Pr(w_i = m - 2 | \{p = 1\} \cap \{b_i = 1\}) \\ & \Pr(w_i \leq m - 4 | \{p = 1\} \cap \{b_i = 1\}) \end{aligned}$$

Let  $\lambda$  be any element of  $\{0, \dots, m^*\}$ .

$$\begin{aligned} \Pr(w_i = \lambda | \{p = 1\} \cap \{b_i = 1\}) &= \frac{\Pr(\{w_i = \lambda\} \cap \{\bigoplus_{k=1}^{m^*} b_k = 1\} \cap \{b_i = 1\})}{\Pr(\{\bigoplus_{k=1}^{m^*} b_k = 1\} \cap \{b_i = 1\})} \\ &= \frac{\Pr(\{w_i = \lambda\} \cap \{w_i \text{ is even}\})}{\Pr(w_i \text{ is even})} \\ &= \begin{cases} 0 & \text{if } \lambda \text{ is odd} \\ \frac{\Pr(\{w_i = \lambda\})}{\Pr(w_i \text{ is even})} & \text{if } \lambda \text{ is even} \end{cases} \end{aligned}$$

We can now compute the probabilistic values we need.

$$\Pr(w_i \geq m | \{p = 1\} \cap \{b_i = 1\}) = \frac{\sum_{\substack{\lambda = m \\ \lambda \text{ even}}}^{m^* - 1} \Pr(w_i = \lambda)}{\Pr(w_i \text{ is even})}$$

$$\Pr(w_i \leq m - 4 | \{p = 1\} \cap \{b_i = 1\}) = \frac{\sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 4} \Pr(w_i = \lambda)}{\Pr(w_i \text{ is even})}$$

Based on this analysis,  $P_i$  is **not** to cheat if:

$$u_i^+ \Pr(w_i = m - 2) + u_i \sum_{\substack{\lambda = m \\ \lambda \text{ even}}}^{m^* - 1} \Pr(w_i = \lambda) + u_i^- \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 4} \Pr(w_i = \lambda) \leq u_i \Pr(w_i \text{ is even})$$

The previous inequality is equivalent to:

$$u_i^+ \alpha^{m-2} (1 - \alpha)^{m^* - (m-1)} \binom{m^* - 1}{m - 2} + u_i^- \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 4} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda+1)} \binom{m^* - 1}{\lambda} \leq u_i \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 1} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda+1)} \binom{m^* - 1}{\lambda}$$

Since  $m$  is even, the sum on the right hand side ends when  $\lambda = m - 2$ . We get:

$$(u_i^+ - u_i) \alpha^{m-2} (1 - \alpha)^{m^* - (m-1)} \binom{m^* - 1}{m - 2} \leq (u_i - u_i^-) \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 4} \alpha^\lambda (1 - \alpha)^{m^* - (\lambda+1)} \binom{m^* - 1}{\lambda}$$

We divide both sides of the previous inequality by  $\alpha^{m-2} (1 - \alpha)^{m^* - (m-1)} (u_i - u_i^-)$ . Defining  $\mathcal{A} := \frac{1 - \alpha}{\alpha}$ , we obtain:

$$\frac{u_i^+ - u_i}{u_i - u_i^-} \binom{m^* - 1}{m - 2} \leq \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m - 4} \binom{m^* - 1}{\lambda} \mathcal{A}^{m - (\lambda+2)} \tag{A.4}$$

The right hand side of Inequality (A.4) is a polynomial of degree  $m - 2$  in  $\mathcal{A}$  with a positive leading coefficient. Since  $\mathcal{A} \xrightarrow{\alpha \rightarrow 0_+} +\infty$ , we deduce that there exists a value  $\alpha_{i,2}$  such that for all  $\alpha \leq \alpha_{i,2}$ , Inequality (A.4) does not hold. In such a situation,  $P_i$  does **not** cheat as indicated in Case 2.

**Case 3.** The current case corresponds to Case 1 of Stage 2 where  $P_i$  sends a fake share. We are essentially in the same situation as in Case 2. Given  $m$  is even, we have:  $w_i \geq m$  (everybody recovers  $s$ ) or  $w_i \leq m - 4$  (nobody recovers  $s$ ) or  $w_i = m - 2$ . In the latter subcase,  $P_i$  will recover  $s$  while the other players will get a wrong value with large probability. The expected payoff of  $P_i$  is at most  $u_i^+$  in that specific subcase. As a consequence, if Inequality (A.4) holds (i.e. we set  $\alpha_{i,3} := \alpha_{i,2}$ ) then  $P_i$  does **not** cheat as indicated in Case 3.

At this point of our proof, we showed that for any  $\alpha \leq \alpha_i^* := \min(\alpha_{i,1}, \alpha_{i,2})$ , player  $P_i$  will follow the protocol's instructions until step 1 of Stage 3 included. Since steps 2.a, 3 and 4 of Stage 3 are run independently by each player, the only remaining way for  $P_i$  to deviate would occur when step 2.b is executed. In the check phase, each single deviation will cause the protocol to stop without anybody learning the secret while, in the renewal phase, a single deviation either may cause every player to recover a wrong secret or may cause the protocol to stop with nobody learning  $s$ . Hence, in both cases, no single rational player  $P_i$  has any incentive to deviate during step 2.b.

As a consequence, for any  $\alpha \leq \alpha^* := \min(\alpha_1^*, \dots, \alpha_n^*)$ , the mechanism  $\Pi(\alpha)$  is a Nash equilibrium. Using the same argument as the proof in Theorem 3.2 from [6], we could demonstrate that  $\Pi(\alpha)$  survives iterated deletion of weakly-dominated strategies.

**Appendix B. Proof of Theorem 19**

Our goal is to show that this choice of  $\tilde{\alpha}$  implies that, for any  $\alpha \leq \tilde{\alpha}$ , both Inequality (A.3) and Inequality (A.4) hold. Indeed, when this occurs, the proof of Theorem 17 exposed in Appendix A indicates that the protocol  $\Pi(\alpha)$  induces a Nash equilibrium surviving iterated deletion of weakly dominated strategies.

Assume that  $\alpha \leq \tilde{\alpha}$ . This inequality implies:

$$\beta \left( \frac{m^* - m + 3}{m - 3} \right)^2 \leq \mathcal{A}^2 \tag{B.1}$$

This inequality leads to:

$$\beta \frac{(m^* - m + 2)(m^* - m + 1)}{(m - 2)(m - 1)} \leq \mathcal{A}^2$$

Therefore, we get:  $\beta \binom{m^*-1}{m-1} \leq \binom{m^*-1}{m-3} \mathcal{A}^2$  and we deduce:

$$\frac{u_i^+ - u_i}{u_i - u_i^-} \binom{m^* - 1}{m - 1} \leq \sum_{\substack{\lambda = 0 \\ \lambda \text{ odd}}}^{m-3} \binom{m^* - 1}{\lambda} \mathcal{A}^{m-(\lambda+1)}$$

which means that Inequality (A.3) is verified.

Let's start from Inequality (B.1) again. It also implies:

$$\beta \frac{(m^* - m + 3)(m^* - m + 2)}{(m - 2)(m - 3)} \leq \mathcal{A}^2$$

Therefore, we get:  $\beta \binom{m^*-1}{m-2} \leq \binom{m^*-1}{m-4} \mathcal{A}^2$  and we deduce:

$$\frac{u_i^+ - u_i}{u_i - u_i^-} \binom{m^* - 1}{m - 2} \leq \sum_{\substack{\lambda = 0 \\ \lambda \text{ even}}}^{m-4} \binom{m^* - 1}{\lambda} \mathcal{A}^{m-(\lambda+2)}$$

which means that Inequality (A.4) is verified which ends our demonstration since this result is valid for every player  $P_i$  ( $i \in \{1, \dots, n\}$ ).