Unconditionally Secure Rational Secret Sharing in Standard Communication Networks

Zhifang Zhang[1,2] and Mulan Liu[1]

Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, CAS, Beijing, China
zfz@amss.ac.cn

Abstract. Rational secret sharing protocols in both the two-party and multi-party settings are proposed. These protocols are built in standard communication networks and with unconditional security. Namely, the protocols run over standard point-to-point networks without requiring physical assumptions or simultaneous channels, and even a computationally unbounded player cannot gain more than ϵ by deviating from the protocol. More precisely, for the 2-out-of-2 protocol the ϵ is a negligible function in the size of the secret, which is caused by the information-theoretic MACs used for authentication. The t-out-of-n protocol is $(t-1)$-resilient and the ϵ is exponentially small in the number of participants. Although secret recovery cannot be guaranteed in this setting, a participant can at least reduce the Shannon entropy of the secret to less than 1 after the protocol. When the secret-domain is large, every rational player has great incentive to participate in the protocol.

Keywords rational secret sharing, ϵ-Nash equilibrium, unconditional security

1 Introduction

Secret sharing [2, 18] is an important tool in cryptography. The widely used t-out-of-n scheme is that a dealer holding a secret distributes shares among n players such that any group of t or more players can recover the secret from their shares while any group of fewer than t players can not. In 2004 Halpern and Teague [8] studied the problem in a game theoretic sense and proposed *rational secret sharing* which is to fulfill the task among rational players who only act in their own self-interest. As Halpern and Teague pointed out that no rational player would broadcast his share in a deterministic recovering process, since keeping silence can guarantee him a utility that is equal to and sometimes even higher than the utilities of other players (because he might be the only one who gets the secret). Therefore most previous secret sharing schemes fail in the rational setting which requires to design a protocol such that all rational players have the incentive for participation. Furthermore, it is more desirable to design a protocol where no player has an incentive to deviate as long as the other players follow the protocol. This requirement is captured by the notion of *equilibrium* in game theory. Although many rational secret sharing schemes [1, 14, 12, 13, 15, 5, 7–11, 20] have been developed achieving kinds of equilibria, they are less satisfactory in some of the following aspects:

Notions of equilibria. Halpern and Teague [8] first proposed achieving a Nash equilibrium *surviving iterated deletion of weakly dominated strategies.* But Kol and Naor [10] later pointed out that some "intuitively bad" strategies cannot be deleted anyway, then they proposed the notion of *strict Nash equilibrium* requiring that each player's strategy is his unique best response to other players' strategies. Although strict Nash equilibrium is a more appealing notion, it is too restrictive to be achieved in many cases. Kol and Naor only achieved strict Nash equilibrium in the two-party case assuming the existence of simultaneous broadcast channels [1]. In non-simultaneous channels, only an approximate equilibrium (i.e. *ϵ-Nash equilibrium*) was achieved. Recently, Fuchsbauer et al. [5] proposed *computational strict Nash equilibrium* and *computational Nash equilibrium that is stable with respect to trembles.* Efficient schemes achieving these equilibria were built in standard communication networks, but only computational security was guaranteed during the protocols. Moreover, equilibria concerning about sequential rationality, such as *everlasting Nash equilibrium* [10] and *sequential equilibrium* [20], were also achieved in the simultaneous channel.

Communication models. Halpern and Teague [8] first assumed private channels, the simultaneous broadcast channel as well as an on-line dealer. Gordon and Katz [7] removed the on-line dealer by using a secure multi-party computation protocol among players, but the simultaneous broadcast channel was still necessary. Actually, many rational secret sharing protocols [1, 14, 15, 20] rely on the assumption of simultaneous channels. Besides, some protocols [9, 12, 13] use even stronger assumptions such as secure envelopes and ballot boxes.

Coalition-resilience. The main drawback of Kol and Naor's construction [10] is that it cannot resist the collusion attack of even two players. But coalition-resilience is an important requirement for t-out-of-n secret sharing. Previous protocols achieved good resilience in either simultaneous broadcast channels [1] or in the computational setting [5, 11].

Unconditional/computational security. In the computational setting, equilibria with good properties (e.g. coalition-resilience [11]) could be achieved and more efficient protocols could run in standard communication networks [5], but it works in the condition that all players are computationally bounded. When higher security is required or players' computing power is unclear, a rational secret sharing protocol with unconditional security (i.e., in the information theoretic setting, such as [10]) is more reliable.

It can see that there is a tradeoff between the above aspects. In this work we focus on rational secret sharing that is coalition-resilient in the information theoretic setting and standard communication networks, at the cost of achieving

[1] When using simultaneous broadcast channels, players must decide on what value (if any) to broadcast in a given round before observing the values broadcast by other players.

ϵ-Nash equilibria only. But we will see that the "ϵ" is quite small and mostly acceptable.

1.1 Our Results and Main Ideas

We first design a 2-out-of-2 rational secret sharing protocol with unconditional security in standard communication networks. The main idea is distributing to player P_1 (resp. P_2) a list of length l_1 (resp. l_2) where $l_2 \leq l_1 \leq l_2 + 1$. Each cell of the lists contains a value, and all the values jointly determine the secret. The recovering phase consists of at most $l_1 + 1$ iterations. In each iteration, say, the j-th iteration, P_1 first broadcasts the value in his j-th cell, then P_2 does similarly. Since the two cases $l_1 = l_2 + 1$ and $l_1 = l_2$ both are possible, P_1 and P_2 cannot know which case really happens before the protocol ends. Therefore each player still has an incentive to broadcast the value even if it comes to his last cell. This protocol achieves an ϵ-Nash equilibrium, where ϵ is a negligible function in the size of the secret and is caused by the information-theoretic MACs used inside.

Then we build a t-out-of-n rational secret sharing protocol that is $(t-1)$-resilient. Since in the information theoretic setting with non-simultaneous channels, a coalition of $t-1$ players can easily get the secret earlier than other players and leave the protocol early, we try to insure that the innocent players (i.e. players who follow the protocol) get as much information as possible. The main idea is to divide each cell into two parts where two values are stored respectively, and the two values are both possible to be the secret if the secret appears in this cell. In each iteration, players first broadcast the first part of the current cell in some order, then the second part. The index indicating whether the current value is the secret or not is to be revealed only after the next value has been recovered. More precisely, suppose the secret appears in the j-th cell which contains s_j^0 and s_j^1 respectively in the two parts. Even the players in a $(t-1)$-coalition at most know that $\mathsf{Prob}[s = s_j^0] = q$ and $\mathsf{Prob}[s = s_j^1] = 1 - q$ for some constant q before seeing the index I_j^1 (i.e. $I_j^1 = 0$ if $s = s_j^1$, and $I_j^1 = 1$ if $s = s_j^0$). But I_j^1 is to be revealed only after recovering s_j^1 (by that time s_j^0 has already been recovered). Therefore after the coalition determines the secret s and leaves the protocol, the rest players at least know $s = s_j^0$ or s_j^1, which is also a pleasant result when the secret-domain is large. On the other hand, the extra gain of the deviating coalition is at most ϵ, where ϵ is exponentially small in the number of participants in the recovering process.

1.2 Related Work

Table 1 displays comparisons in some aspects between our protocols in this paper and those in some previous work.

Kol and Naor [10] provided constructions in both simultaneous and non-simultaneous channels in the information theoretic setting. Our constructions are similar to theirs in that shares are both in the form of lists with different

	equilibrium	channel	coalition resilience	security
KN-[10]	strict Nash	simultaneous	1-resilient	unconditional
	ϵ-Nash	non-simultaneous	1-resilient	unconditional
ADGH-[1]	ϵ-Nash	simultaneous	k-resilient	computational/ unconditional
FKN-[5]	strict Nash	non-simultaneous	$(t-1)$-resilient	computational
This paper	ϵ-Nash	non-simultaneous	$(t-1)$-resilient	unconditional

Table 1.

length and the recovering is accomplished by revealing the lists cell by cell. But our 2-out-of-2 protocol is more efficient because shorter lists are involved and simpler cells are contained. Details will be found in the remarks after Theorem 1. General k-resilience was discussed in [1] where it achieved unconditional security for $k < \frac{n}{3}$ and computational security for $k < n$. But the protocols in [1] relied on simultaneous channels. Efficient protocols with optimal coalition resilience in standard communication networks were designed in [5]. Most importantly, it achieved equilibria with appealing properties, such as strict Nash, and stability with respect to trembles. But only computational security was guaranteed from the beginning of the recovering process.

2 Preliminaries

In this section it introduces notions about rational secret sharing and information-theoretic MACs, as well as concepts of the equilibrium to be achieved in this work.

2.1 Secret Sharing and Players' Utilities

In a t-out-of-n secret sharing scheme, a dealer (denoted as Dealer hereafter) holding a secret distributes shares among n players such that the following two conditions are satisfied:

1. **Recoverability.** Any group of t or more players puting their shares together can uniquely determine the secret.
2. **Secrecy.** Any group of fewer than t players cannot recover the secret.

It usually assumes that Dealer is the trusted third party and each player is either honest or malicious. In a game theoretic view, it is more realistic to view each player as a rational party who acts only in his interest. To model rationality, we define for each player P_i a real-valued utility function u_i such that everyone's interest is to maximize his utility. The commonly used assumptions for defining utilities in rational secret sharing are as follows [8]:

- Each player always prefers to learn the secret than to not learn it;

- Secondarily, each player prefers that the fewer of the other players who get it, the better.

In particular, we define four utility values for each player P_i :

(1) $u_i = a$ if P_i gets the secret while P_j does not for any $j \neq i$;
(2) $u_i = b$ if P_i gets the secret and so does P_j for some $j \neq i$;
(3) $u_i = c$ if P_i does not get the secret and neither does P_j for any $j \neq i$;
(4) $u_i = d$ if P_i does not get the secret while P_j does for some $j \neq i$.

From the common assumptions on utilities, it obviously holds that $a > b > c > d$. Let S denote the secret-domain and $|S|$ be the cardinality of S. Then by guessing the secret uniformly from S, a player at most gets the utility

$$U_{random} = \frac{1}{|S|}a + (1 - \frac{1}{|S|})c \; .$$

To make every player has the incentive to participate in a protocol for secret recovering, it requires $b > U_{random}$.

Concerning about coalitions, for simplicity we additionally assume that

- Once a player joins a coalition, he will never leave the coalition before the protocol ends;
- Players in the same coalition always share all information they jointly have.

Given an execution of a protocol, let $\mathcal{C}(i)$ denote the coalition that P_i joined in. Thus all players in $\mathcal{C}(i)$ have the same utility as P_i. As an extension, we similarly define the four utility values a, b, c, d for each player P_i as in (1)-(4) just replacing "$j \neq i$" with "$j \notin \mathcal{C}(i)$".

When no coalition is formed, namely, $\mathcal{C}(i) = \{i\}$ for any $i \in \{1, ..., n\}$, the problem is much easier [10]. In this work we deal with the most general coalitions in t-out-of n secret sharing, i.e. $1 \leq |\mathcal{C}(i)| \leq t - 1$.

2.2 Notions of Equilibria

In the recovering process of a secret sharing scheme, view the interaction between players as a game among the n players. Let $\sigma = (\sigma_1, ..., \sigma_n)$ denote a strategy profile of players, where σ_i is P_i's strategy for $1 \leq i \leq n$. Usually, we let σ_{-i} denote the strategy profile of all players except P_i and $\sigma_{\mathcal{C}}$ denote the strategy profile constricted to the coalition $\mathcal{C} \subseteq \{1, ..., n\}$. Given a strategy profile σ, it induces the utility $u_i(\sigma)$ for each player P_i. Referring to the definitions in [1, 5, 10, 11], we give some notions of equilibria as follows:

Definition 1. *A strategy σ induces an ϵ-Nash equilibrium if for any player P_i and any strategy σ'_i of P_i, it holds that*

$$u_i(\sigma'_i, \sigma_{-i}) \leq u_i(\sigma_i, \sigma_{-i}) + \epsilon \; .$$

When $\epsilon = 0$ it is the well-known Nash equilibrium [16]. In some cases, a Nash equilibrium in the strict sense is hard to compute [3], while computing the ϵ-approximate Nash equilibrium is much easier [4]. Therefore, the ϵ-Nash equilibrium is also an appealing notion for a small ϵ.

Definition 2. *A strategy σ induces an k-resilient ϵ-Nash equilibrium if for any coalition $\mathcal{C}$ of at most k players (i.e. $|\mathcal{C}| \leq k$) and for any strategy profile $\sigma'_{\mathcal{C}}$ of the coalition $\mathcal{C}$, it holds that*

$$u_i(\sigma'_{\mathcal{C}}, \sigma_{\overline{\mathcal{C}}}) \leq u_i(\sigma_{\mathcal{C}}, \sigma_{\overline{\mathcal{C}}}) + \epsilon \quad \textit{for any} \;\; i \in \mathcal{C} \ ,$$

where $\overline{\mathcal{C}}$ denotes the complement of $\mathcal{C}$.

When $k = 1$ it is the ϵ-Nash equilibrium just defined. In this work, we realize the resilience for $k = t - 1$ in a t-out-of-n secret sharing scheme. Obviously, this is the optimal coalition resilience in the t-out-of-n case.

2.3 Information-Theoretic MACs

We refer to [6] for the description of information theoretically secure message authentication codes (MACs). A message authentication code consists of three polynomial-time algorithms (Gen,Mac,Vrfy). The key-generation algorithm Gen takes as input the security parameter 1^m and outputs a key k. The message authentication algorithm Mac takes as input a key k and a message $M \in \{0,1\}^{\leq m}$, and outputs a tag t; we write this as $t = \mathsf{Mac}_k(M)$. The verification algorithm Vrfy takes as input a key k, a message M and a tag t, and outputs a bit b; i.e., $b = \mathsf{Vrfy}_k(M, t)$. We regard $b = 1$ as acceptance and $b = 0$ as rejection, and require that for all m, all k output by $\mathsf{Gen}(1^m)$, all $M \in \{0,1\}^{\leq m}$, it holds that $\mathsf{Vrfy}_k(M, \mathsf{Mac}_k(M)) = 1$.

Definition 3. *(Gen,Mac,Vrfy) is an information-theoretic MAC if for any $M \in \{0,1\}^{\leq m}$, $k = \mathsf{Gen}(1^m)$, $t = \mathsf{Mac}_k(M)$, and for any (computationally unbounded) adversary $\mathcal{A}$, the following probability is negligible in m:*

$$\mu(m) = \mathsf{Prob}\ [(M', t') \leftarrow \mathcal{A}(M, t) : \mathsf{Vrfy}_k(M', t') = 1 \bigwedge M' \neq M] \ .$$

For example, an information-theoretic MAC can be built as follows [17, 19]: Let $\mathbb{F}$ be a finite field, the key is $(\alpha, \beta) \in \mathbb{F}^2$. For a message $M \in \mathbb{F}$, the tag is generated as $t = \beta - \alpha M \in \mathbb{F}$.

3 Rational Secret Sharing: The 2-Out-of-2 Case

In this section we give a 2-out-of-2 rational secret sharing protocol in standard communication networks (i.e. point-to-point and non-simultaneous channel) and with unconditional security. Denote the protocol by $\mathit{\Pi}$, we describe $\mathit{\Pi}$ in terms of Dealer's protocol and players' protocol separately. Actually, Dealer's protocol corresponds to the distributing phase, and players' protocol corresponds to the recovering phase where only players are active.

Let $S = \{0,1\}^m$ be the secret-domain and $s \in S$ be the secret. For player P_1 and P_2, let a, b, c, d be the utility values as defined in Section 2.1. Suppose $(\mathsf{Gen},\mathsf{Mac},\mathsf{Vrfy})$ is an information-theoretic MAC.

Dealer's Protocol.

1. Choose an integer $l \in \mathbb{N}$ according to a geometric distribution with parameter p [2], where p is a constant to be determined later (in Theorem 1).

2. Determine the two integers l_1 and l_2 such that $l_1 + l_2 = l + 1$ and $l_2 \leq l_1 \leq l_2 + 1$.

3. Randomly select $a_1, ..., a_{l_1} \in S$ and $b_1, ..., b_{l_2} \in S$ such that

$$(\oplus_{i=1}^{l_1} a_i) \oplus (\oplus_{i=1}^{l_2} b_i) = s \ .$$

4. Generate secret keys $\alpha_1, ..., \alpha_{l_2+1}$ and $\beta_1, ..., \beta_{l_1}$ for the MAC by $\mathsf{Gen}(1^m)$. Construct two lists L_1 and L_2 of length l_1 and l_2 respectively, where for $1 \leq i \leq l_1$ (resp. $1 \leq i \leq l_2$) the i-th cell of L_1 (resp. L_2) contains a_i, $\mathsf{Mac}_{\alpha_i}(a_i)$ and β_{i-1} (resp. contains b_i, $\mathsf{Mac}_{\beta_i}(b_i)$ and α_i).

5. Send the list L_1 and the secret key β_{l_1} (resp. the list L_2 and the secret key α_{l_2+1}) to P_1 (resp. P_2).

Players' Protocol.

It consists of l_1 or $l_1 + 1$ iterations. For $1 \leq j \leq l_1 + 1$, the j-th iteration goes along the following two rounds:

1. Denote by $(b'_{j-1}, t^{(b)}_{j-1})$ the message that P_1 received from P_2 in last round. Player P_1 first checks if it holds $\mathsf{Mac}_{\beta_{j-1}}(b'_{j-1}) = t^{(b)}_{j-1}$ (Note for $j = 1$ this check is not needed). If it holds, then P_1 sends $(a_j, \mathsf{Mac}_{\alpha_j}(a_j))$ to P_2; otherwise, P_1 quits and outputs $(\oplus_{i=1}^{j-1} a_i) \oplus (\oplus_{i=1}^{j-2} b'_i)$ as the secret.

2. Denote by $(a'_j, t^{(a)}_j)$ the message that P_2 received from P_1 in last round. Player P_2 checks if it holds $\mathsf{Mac}_{\alpha_j}(a'_j) = t^{(a)}_j$. If it holds, P_2 sends $(b_j, \mathsf{Mac}_{\beta_j}(b_j))$ to P_1; otherwise, P_2 quits and outputs $(\oplus_{i=1}^{j-1} a'_i) \oplus (\oplus_{i=1}^{j-1} b_i)$ as the secret.

If a player's list comes to the end, i.e., the j-th cell of his list is empty, then after verifying the message just received from the opposite, he sends the message "end" in the j-th iteration. After that both players stop running and set the secret to be the XOR of all the values revealed so far.

In brief, the recovering process is accomplished by letting the two players alternately reveal their lists cell by cell, while P_1 goes first. Figure 1 describes the recovering process when $l_1 = l_2$.

Then we give some intuition as to why the recovering process of $\mathit{\Pi}$ (i.e. players' protocol) is an ϵ-Nash equilibrium for an appropriate choice of p, where $\epsilon = \epsilon(m)$ is a negligible function in length of the secret.

[2] Suppose in each coin toss, the Head appears with probability p. Then l is the number of independent tosses needed until the first Head turns up.

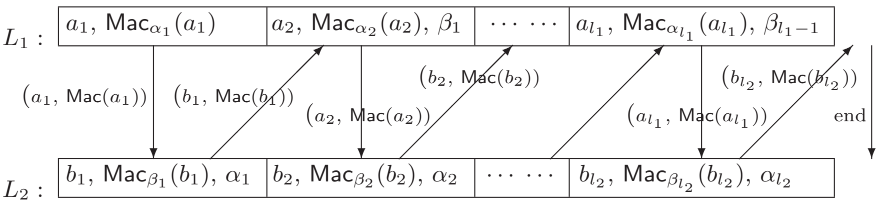
L_1 :
a_1, $\mathsf{Mac}_{\alpha_1}(a_1)$
a_2, $\mathsf{Mac}_{\alpha_2}(a_2)$, β_1
$\cdots\cdots$
a_{l_1}, $\mathsf{Mac}_{\alpha_{l_1}}(a_{l_1})$, β_{l_1-1}
$(a_1, \mathsf{Mac}(a_1))$
$(b_1, \mathsf{Mac}(b_1))$
$(a_2, \mathsf{Mac}(a_2))$
$(b_2, \mathsf{Mac}(b_2))$
$(a_{l_1}, \mathsf{Mac}(a_{l_1}))$
$(b_{l_2}, \mathsf{Mac}(b_{l_2}))$
end
L_2 :
b_1, $\mathsf{Mac}_{\beta_1}(b_1)$, α_1
b_2, $\mathsf{Mac}_{\beta_2}(b_2)$, α_2
$\cdots\cdots$
b_{l_2}, $\mathsf{Mac}_{\beta_{l_2}}(b_{l_2})$, α_{l_2}

Fig. 1. The recovering process when $l_1 = l_2$.

(a) P_1 has no incentive to deviate in the first iteration.
Since $l_1 + l_2 = l + 1 > 1$, it must have $l_2 \geq 1$. Namely, P_2 at least holds a value that contributes to determining s. P_1 cannot get this value if his message broadcast in the first iteration does not pass verification of the MAC. So by deviating, P_1 can get utility at most $\mu(m)a + (1 - \mu(m))U_{random}$, where $\mu(m)$ is the probability of successfully forging an MAC as defined in Definition 3 and $U_{random} = \frac{1}{|S|}a + (1 - \frac{1}{|S|})c$ is an upperbound of the utility that a player can get by guessing the secret uniformly from S. By requiring

$$\mu(m)a + (1 - \mu(m))U_{random} < b \tag{1}$$

P_1 has no incentive to deviate in this iteration.

(b) For $2 \leq j \leq l_1$, P_1 has no incentive to deviate in the j-th iteration.
Similarly to the analysis in (a), P_1 has no incentive to deviate through iteration 2 to $l_1 - 1$. Achieving the l_1-th iteration, with probability p it holds that $l_2 = l_1 - 1$, i.e. P_2's list has run out. In this situation, P_1 can get utility at most a by deviation. But if $l_2 = l_1$ which happens with probability $1 - p$, P_1 get at most $\mu(m)a + (1 - \mu(m))U_{random}$. Therefore P_1 will not deviate by requiring

$$pa + (1 - p)(\mu(m)a + (1 - \mu(m))U_{random}) < b\ . \tag{2}$$

Note that inequality (2) implies inequality (1).

(c) For $1 \leq j \leq l_2$, P_2 has no incentive to deviate in the j-th iteration.
The analysis is similar to that of (b).

(d) P_1 (resp. P_2) cannot increase his utility more than ϵ by deviating in the $(l_1 + 1)$-th (resp. the $(l_2 + 1)$-th) iteration.
In the $(l_1 + 1)$-th iteration and after verifying the MAC, P_1 already knows that $l_2 = l_1$ and he can determine $s = (\oplus_{i=1}^{l_1} a_i) \oplus (\oplus_{i=1}^{l_2} b'_i)$. But P_2 still does not know whether P_1's list is longer than his or not. P_1 can deceive P_2 by continuing to send a fake value in the $(l_1 + 1)$-th iteration which passes verification of the MAC under the secret key $\alpha_{l_1+1} = \alpha_{l_2+1}$, and the success probability is at most $\mu(m)$ due to security of the MAC. Thus P_1 can get utility at most $\mu(m)a + (1 - \mu(m))b$. Therefore,

$$\epsilon(m) = \mu(m)a + (1 - \mu(m))b - b = \mu(m)(a - b)\ .$$

The analysis of P_2's $(l_2 + 1)$-th iteration is similar.

From the analysis (a)-(d), it immediately has the following theorem.

Theorem 1. *If the parameter p satisfies the inequality (2), then the protocol Π for 2-out-of-2 rational secret sharing induces an ϵ-Nash equilibrium with $\epsilon = \mu(m)(a - b)$, where $\mu(m)$ is the negligible probability of successfully forging an information-theoretic MAC.*

Remark 1. The 2-out-of-2 protocol in [10] used lists of length $l' - 1$ and $l' + d' - 1$ respectively, where l' and d' both were chosen according to a geometric distribution with parameter β. Our protocol Π uses lists of length l_1 and l_2 respectively where $l_1 + l_2 - 1$ is chosen according to a geometric distribution with parameter p. Since both β and p are determined by the utility values under the similar inequalities, we can simply regard $\beta = p$. Then the expected length of lists in [10] are $\frac{1}{p} - 1$ and $\frac{2}{p} - 1$, while our lists are both of length about $\frac{1}{2p}$. That is, we only need the list that is almost half as long as the shorter list in [10], which means the expected size of shares in our protocol is smaller.

Remark 2. Since in [10] the shorter list was just a prefix of the longer one and every value alone could possibly be the secret, a player can certainly determine the secret if he finds all his remain cells contain the same value. To fix this problem, it masked each value by a random number for each cell. Thus the cells in [10] contained both the masked value and share of the mask. But in our protocol, the secret is jointly determined by all values contained in the two lists, a player cannot determine the secret even if he sees all values in his list. Therefor no mask is needed in our protocol and our lists consist of simpler cells.

4 Rational Secret Sharing: The t-Out-of-n Case

We now construct a t-out-of-n rational secret sharing protocol in the information theoretic setting. Since it is in non-simultaneous channels and $(t - 1)$-resilience is required, the protocol is not a simple extension of the protocol Π constructed in Section 3. Denote the t-out-of-n protocol by Π'. We still describe Π' in terms of Dealer's protocol and players' protocol separately.

Dealer's Protocol.

1. Choose integers l^* and d according to a geometric distribution with parameter p', where p' is a constant to be determined later (in Theorem 2).

2. Randomly select $\sigma \in \{0, 1\}$ such that $\mathsf{Prob}[\sigma = 0] = q$, where q is a constant to be determined later (in Theorem 2).

3. Construct a list of length $l = l^* + d$. For $1 \leq j \leq l$, the j-th cell contains:

- Main: $(s_j^0, s_j^1) \in S^2$, where S is the secret-domain. In particular, it requires $s_{l^*}^{\sigma} = s$ and the other values are randomly chosen.

- Index: $(I_j^0, I_j^1) \in \{0,1\}^2$ where

$$I_j^0 = \begin{cases} 1, & \text{if } j-1 = l^* \text{ and } \sigma = 1 \\ 0, & \text{otherwise} \end{cases}, I_j^1 = \begin{cases} 1, & \text{if } j = l^* \text{ and } \sigma = 0 \\ 0, & \text{otherwise} \end{cases}.$$

  For consistence, fix $I_1^0 = 0$.
- Permutation: $\pi_j \in \mathit{\Pi}_n$ where $\mathit{\Pi}_n$ denotes the set of all permutations on $\{1, ..., n\}$ [3].

4. Randomly select a permutation $\pi_0 \in \mathit{\Pi}_n$, and send π_0 to all players.

5. Suppose $i_0 \in \{1, ..., n\}$ appears first in the permutation π_{l^*-1}. Construct n lists, denoted by $L_1, ..., L_n$, where L_{i_0} is of length l^* and the other $n-1$ lists are of length l. For $1 \le i \le n$ and $1 \le j \le l$, the j-th cell of L_i contains: (Note the list L_{i_0} ends after the l^*-th cell)

- Share of main: s_{ji}^0 and s_{ji}^1, where s_{ji}^0 (resp. s_{ji}^1) is a (t, n)-share [4] of s_j^0 (resp. s_j^1).
- Share of index: I_{ji}^0 and I_{ji}^1, where I_{ji}^0 (resp. I_{ji}^1) is a (t, n)-share of I_j^0 (resp. I_j^1).
- Share of permutation: π_{ji} which is a (t, n)-share of π_j.
- Authentication information: The tags

$$\{\mathsf{Mac}_{\alpha_{j,i,h}}(s_{ji}^0), \mathsf{Mac}_{\alpha'_{j,i,h}}(s_{ji}^1), \mathsf{Mac}_{\beta_{j,i,h}}(I_{ji}^0), \mathsf{Mac}_{\beta'_{j,i,h}}(I_{ji}^1), \mathsf{Mac}_{\gamma_{j,i,h}}(\pi_{ji}) \mid \substack{1 \le h \le n, \\ h \ne i}\}$$

  and the keys $\{\alpha_{j,h,i}, \alpha'_{j,h,i}, \beta_{j,h,i}, \beta'_{j,h,i}, \gamma_{j,h,i} \mid 1 \le h \le n, h \ne i\}$. We note that the key $\alpha_{j,h,i}$ is used to verify a tag of s_{jh}^0 and is stored in the j-th cell of L_i.

6. For $1 \le i \le n$, send the list L_i to player P_i.

Players' Protocol.

Suppose k $(k \ge t)$ players are to jointly recover the secret. The recovering process consists of at most l iterations. In the j-th iteration for $1 \le j \le l$, if the protocol does not end, the players do the following:

1. Recover s_j^0. In the order determined by the permutation π_{j-1}, each player (say, P_i) sends to the other players $(s_{ji}^0, \mathsf{Mac}(s_{ji}^0))$. Hereafter we usually omit the key in the MAC because it is clearly determined by the message and the receiver. Players verify the MACs after receiving messages. If all messages pass the verification, then each player recovers s_j^0.

2. Recover I_j^0. Still in the order of π_{j-1} players send their shares along with MACs, and then recover I_j^0.

3. Recover s_j^1. Same as above.

4. Recover I_j^1. Same as above.

[3] Precisely, the permutation π_j denotes an order in which players send messages in the $(j+1)$-th iteration.

[4] The share can be generated by Shamir's (t, n)-threshold secret sharing scheme.

5. Recover π_j. Same as above.

In any of the above five steps, a player quits from the protocol at encountering any one of the following situations.

- His list has run out. Then he quits and sets the secret to be the last value he recovered. For example, if his list is of length l' and the protocol does not end after the first l' iterations, then he quits in the $(l'+1)$-th iteration and sets $s = s^1_{l'}$.
- Find some index $I^\delta_j = 1$. Then he quits and sets $s = s^{1-\delta}_{j-1+\delta}$.
- Find someone cheats in recovering s^0_j. Then he quits and sets $s = s^1_{j-1}$.
- Find someone cheats in recovering I^0_j. Then he quits and sets $s = s^1_{j-1}$ with probability $1-q$ and $s = s^0_j$ with probability q.
- Find someone cheats in recovering s^1_j. Then he quits and sets $s = s^0_j$.
- Find someone cheats in recovering I^1_j. Then he quits and sets $s = s^0_j$ with probability q and $s = s^1_j$ with probability $1-q$.
- Find someone cheats in recovering π_j. Then he quits and sets $s = s^1_j$.

Now we give some analysis to explain why the recovering process of Π' induces an ϵ-Nash equilibrium with $(t-1)$-resilience. For simplicity, we neglect the negligible part of ϵ caused by successfully forging the MAC. As a warm-up, we first show that any single player has no incentive to deviate from the protocol. For a single player P_i, there are two cases:

(a) P_i holds a list of length l.
It is important to note that P_i cannot know he is holding the long list until the protocol ends or it comes to his last cell (i.e. the l-th cell). Therefore, for $1 \le j < l$, P_i guesses $l^* = j$ and deviates in the j-th iteration, then he can get utility at most $p'a + (1-p')U_{random}$. P_i has no incentive to deviate if it holds

$$p'a + (1-p')U_{random} < b \ . \tag{3}$$

When it comes to the last cell (i.e. the l-th cell) and P_i is not the first one to send messages according to π_{l-1}, then P_i knows that $l^* = l-1$ and $s = s^1_{l-1}$. Actually, every other player can also conclude $s = s^1_{l-1}$ no matter what P_i does in the l-th iteration. Thus P_i has no incentive to deviate.

(b) P_i holds a list of length l^*.
Similarly, it can see that P_i has no incentive to deviate in the j-th iteration for $1 \le j \le l^*-1$, if the inequality (3) holds. When it comes to the l^*-th iteration P_i knows he is holding the short list because he is the first to send messages in that iteration. Since P_i is the first one to talk in the l^*-th iteration, when P_i determines for sure what the secret is, so do the other players. Thus P_i has no incentive to deviate.

Then we give some intuition as to why the recovering process of Π' is $(t-1)$-resilient. For any coalition $\mathcal{C}$ with $1 < |\mathcal{C}| \le t-1$, there are two cases:

(c) The short list holder is contained in $\mathcal{C}$.

Since the lists are of different length, players in $\mathcal{C}$ can easily determine l^* in advance. Thus ignoring the negligible probability of forging the MAC successfully, the best option for players in $\mathcal{C}$ is to get as much information about $\{s^0_{l^*}, s^1_{l^*}, I^1_{l^*}\}$ as possible and secondarily, to make players outside $\mathcal{C}$ know as little as possible. It is easy to see that if the inequality (3) holds $\mathcal{C}$ has no incentive to deviate before the l^*-th iteration. In the l^*-th iteration,

- If $\mathcal{C}$ deviates in recovering $s^0_{l^*}$, the best result for $\mathcal{C}$ is that they get $s^0_{l^*}$ while no one else does. Thus $\mathcal{C}$ guesses $s = s^0_{l^*}$ and the other players set $s = s^1_{l^*-1}$. Since Dealer set $s = s^0_{l^*}$ with probability q, $\mathcal{C}$ guesses wrong with probability $1-q$. Therefore by deviating players in $\mathcal{C}$ get utility at most $qa + (1-q)c$. Requiring

$$qa + (1-q)c < b \ , \tag{4}$$

then $\mathcal{C}$ has no incentive to deviate.

- When recovering $I^0_{l^*}$, since $I^0_{l^*}$ only indicates whether $s^1_{l^*-1}$ is the secret or not which $\mathcal{C}$ has already known. Besides, at this time players outside $\mathcal{C}$ already get $s^0_{l^*}$ which means they also have opportunity to get the right secret even if $\mathcal{C}$ deviates. Based on the inequality (4), $\mathcal{C}$ has no incentive to deviate.
- If $\mathcal{C}$ deviates in recovering $s^1_{l^*}$, then players in $\mathcal{C}$ set $s = s^0_{l^*}$ with probability q and set $s = s^1_{l^*}$ with probability $1-q$. By the protocol Π', after detecting someone cheats in recovering $s^1_{l^*}$, each of the players outside $\mathcal{C}$ sets $s = s^0_{l^*}$ and quits. If Dealer set $\sigma = 0$ (which happens with probability q), then with probability q all players get the right secret and with probability $1-q$ players in $\mathcal{C}$ guess wrong while others guess right. If Dealer set $\sigma = 1$ (which happens with probability $1-q$), then players outside $\mathcal{C}$ get the wrong secret, while $\mathcal{C}$ guesses right with probability $1-q$.

  Thus deviation in recovering $s^1_{l^*}$ makes players in $\mathcal{C}$ get utility at most

$$q(qb+(1-q)d)+(1-q)(qc+(1-q)a) = (1-q)^2a+q^2b+q(1-q)(c+d) \ .$$

  By requiring

$$(1-q)^2a + q^2b + q(1-q)(c+d) < b \ , \tag{5}$$

  $\mathcal{C}$ has no incentive to deviate.

- If $\mathcal{C}$ deviates in recovering $I^1_{l^*}$, we will show that players in $\mathcal{C}$ can increase the utility by at most $\epsilon = O(\lambda^k)$ where k is the number of participants in the recovering process and $\lambda < 1$ is a constant determined by q. After deviation players in $\mathcal{C}$ can determine the secret, while each player outside $\mathcal{C}$ sets $s = s^0_{l^*}$ with probability q and $s = s^1_{l^*}$ with probability $1-q$. Suppose $|\mathcal{C}| = c$, then there are $k-c$ players outside $\mathcal{C}$. If Dealer set $\sigma = 0$, then the probability that none of the $k-c$ players outputs the right secret is $(1-q)^{k-c}$, while if $\sigma = 1$, this probability is q^{k-c}. Thus by deviation players in $\mathcal{C}$ get utility at most

$$\begin{aligned} U_D &= q((1-q)^{k-c}a + (1-(1-q)^{k-c})b) + (1-q)(q^{k-c}a + (1-q^{k-c})b) \\ &= (q(1-q)^{k-c} + (1-q)q^{k-c})a + (1 - q(1-q)^{k-c} - (1-q)q^{k-c})b \ . \end{aligned}$$

Therefore $\epsilon = U_D - b = (q(1-q)^{k-c} + (1-q)q^{k-c})(a-b)$. Denote $\lambda = max\{q, 1-q\}$, then $\epsilon \leq \lambda^{k-c}(a-b) = O(\lambda^k)$.

- Neglecting the negligible probability of successfully forging a MAC, $\mathcal{C}$ has no incentive to deviate after recovering $I^1_{l^*}$, because $\mathcal{C}$ has already known the secret and players outside $\mathcal{C}$ can also output the right secret.

(d) The short list holder is not contained in $\mathcal{C}$.
Then the coalition $\mathcal{C}$ can only know $l^* \leq l-1$ in advance. By the analysis similar to that of (a), $\mathcal{C}$ has no incentive to deviate in the j-th iteration for $1 \leq j < l-1$. In the $(l-1)$-th iteration, similar to the analysis of the fourth situation in (c), $\mathcal{C}$ can only increase the utility by at most $\lambda^{k-c}(a-b)$ if they deviates from the protocol.

From the analysis (a)-(d) above, we can get the following theorem.

Theorem 2. *Let the parameters p', q and the utility values satisfy the inequalities (3)-(5), then the protocol Π' for t-out-of-n rational secret sharing induces a $(t-1)$-resilient ϵ-Nash equilibrium with $\epsilon < \lambda^{k-t+1}(a-b)$, where $\lambda = max\{q, 1-q\}$ and k is the number of participants in the recovering process.*

Remark 3. Note that the inequality (4) and (5) may not simultaneously hold for some values of a, b, c, d. This can be solved by making some additional assumptions on the utility values. For example, assume that $a-b < b-c$, then the inequality (4) and (5) are satisfied for $\frac{a-b}{a-c} < q < \frac{b-c}{a-c}$. Actually, the assumption $a-b < b-c$ is implied from the natural requirement of $U_{random} < b$ for $|S| = 2$, i.e. each player still has an incentive to participate in the protocol for recovering even if the secret is just one bit.

Remark 4. It can see that the ϵ is exponentially small in the number of participants. When a large number of players participate in the recovering process or the utility values a and b are very close, a coalition of $(t-1)$ players cannot gain much by deviation form Π'. Actually, as pointed out in [10] a gain by a $(t-1)$-coalition is inevitable in the information theoretic setting. We leave it as an open problem to determine the lower bound of ϵ at achieving $(t-1)$-resilience in standard communication networks.

On the other side, although some players quit from the protocol after they get the secret, leaving the other players (who honestly follow the protocol so far, thus we call them "innocent players") cannot determine what the secret is, the innocent player can at least be sure that the secret must be one of the two values he has already recovered. Thus in innocent players' view the Shannon entropy of the secret reduces to less than 1. When $|S|$ is very large, every rational player has great incentive to participate in the protocol Π' even if he might encounter a coalition of $t-1$ players.

5 Conclusions

In the information theoretic setting of rational secret sharing, only approximate Nash equilibrium can be achieved in standard communication networks. We realize ϵ-Nash both for the 2-out-of-2 case and the t-out-of-n case. The 2-out-of-2 protocol is more efficient than previous ones and the ϵ is a negligible function in the size of the secret. This negligible function is due to the information-theoretic MAC used inside. The t-out-of-n protocol is $(t-1)$-resilient and the ϵ is exponentially small in the number of participants. We leave it as an open problem to determine the lower bound of ϵ in both cases.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: 25th ACM Symposium Annual on Principles of Distributed Computing, pp. 53–62. ACM Press, New York (2006)
2. Blakley, G.R. : Safeguarding cryptographic keys. Proceedings of the National Computer Conference, American Federation of Information Processing Societies Proceedings 48: 313–317 (1979)
3. Daskalakis, C., Goldberg, P., Papadimitriou, C. : The complexity of computing a Nash equilibrium. In Proc. STOC 2006, pp. 71–78, ACM Press, (2006)
4. Daskalakis, C., Mehta, A., Papadimitriou, C. : A note on approximate Nash equilibria. International Workshop on Internet and Network Economics, pp. 297–306, (2006)
5. Fuchsbauer, G., Katz, J., Naccache, D. : Efficient Rational Secret Sharing in Standard Communication Networks. TCC 2010, LNCS 5978, pp. 419–436 (2010)
6. Gordon, S.D.,Hazay, C., Katz, J., Lindell,Y. : Complete fairness in secure two-party computation. In Proc. STOC 2008, pp. 413–422, ACM Press, (2008)
7. Gordon, S.D., Katz, J. : Rational secret sharing, revisited. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 229–241. Springer, Heidelberg (2006)
8. Halpern, J., Teague, V. : Rational secret sharing and multiparty computation. In: Proc. of 36th STOC, pages 623–632. ACM Press (2004)
9. Izmalkov, S., Micali, S., Lepinski, M.: Rational secure computation and ideal mechanism design. In: 46th Annual Symposium on Foundations of Computer Science (FOCS), pp. 585–595. IEEE, Los Alamitos (2005)
10. Kol, G., Naor, M. : Games for exchanging information. In: STOC 2008, pp. 423–432. ACM, New York (2008)
11. Kol, G., Naor, M.: Cryptography and game theory: Designing protocols for exchanging information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
12. Lepinski, M., Micali, S., Peikert, C., Shelat, A.: Completely fair SFE and coalition-safe cheap talk. In: 23rd ACM Symposium Annual on Principles of Distributed Computing, pp. 1–10. ACM Press, New York (2004)
13. Lepinski, M., Micali, S., Shelat, A.: Collusion-free protocols. In: 37th Annual ACM Symposium on Theory of Computing (STOC), pp. 543–552. ACM Press, New York (2005)

14. Lysyanskaya, A., Triandopoulos, N.: Rationality and adversarial behavior in multi-party computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
15. Maleka, S., Shareef, A., Rangan, C.P.: The deterministic protocol for rational secret sharing. In: IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2008, pp. 1–7 (2008)
16. Osborne, M., Rubinstein, A. : A Course in Game Theory, MIT Press, Cambridge (2004)
17. Rabin, T., Ben-Or, M. : Verifiable Secret Sharing and Multiparty Protocols with Honest Majority. In Proceedings of the 21th Annual ACM Symposium on Theory of Computing (STOC), pp. 73–85, (1989)
18. Shamir, A. : How to share a secret, Communications of the ACM, 22(11), pp. 612–613 (1979)
19. Wegman, M., Carter, L. : New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences, volume 22, pp. 265–279, (1981)
20. Zhang, Z. : Rational secret sharing as extensive games, Avalable online: http://eprint.iacr.org/2010/184.