An Efficient Rational Secret Sharing Protocol Resisting against Malicious Adversaries over Synchronous Channels

Yang Yu and Zhanfei Zhou

State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
Beijing 100093, P.R. China
{yyu,zhouzhanfei}@is.ac.cn

Abstract. Current works solve the problem of rational secret sharing from one or some, but not all, of the following aspects: achieving a more appealing equilibrium concept, avoiding strong communication models and resisting against adversaries. To address one issue above, they need to lower the satisfaction in other issues. In this paper we construct a t-out-of-n rational secret sharing protocol, which achieves an enhanced notion of *computational strict Nash equilibrium with respect to adversary structure* $\mathcal{A}$, runs over synchronous (non-simultaneous) broadcast channels and tolerates a malicious adversary who controls a minority of players. To the best of our knowledge, compared with current works tolerating adversaries, we are the first to yield positive results in all the three research aspects above. The feasibility of our protocol is based on the use of publicly verifiable secret sharing. Under the assumptions related to discrete logarithm and ElGamal cryptosystem, computational bounded players have an incentive not to deviate no matter how adversaries behave.

Keywords: Rational secret sharing, game theory, malicious adversary, computational strict Nash equilibrium, synchronous channel.

1 Introduction

1.1 Background

Secret sharing studies the problem that the dealer shares a secret among players such that an authorized subset of players can recover the secret but an unauthorized subset of players can get no information about the secret, it is fundamental in the area of secure multiparty computation (SMPC). Rational secret sharing (RSS), proposed by Halpern and Teague in 2004 [7], considers the problem of secret sharing in the game-theoretic model where all players are rational. Different from honest players and malicious players, rational players behave in their interests and try to get the maximum payoffs. It is assumed that they prefer to get the secret and prefer that the fewer players who get the secret

M. Kutyłowski and M. Yung (Eds.): Inscrypt 2012, LNCS 7763, pp. 69–89, 2013.

the better. Since the work of Halpern and Teague, many works have been devoted to rational secret sharing, such as [1,10,9,5,16], they focus on the following aspects.

Equilibrium Standard. The goal of rational secret sharing is to motivate all players to follow the protocol, hence, Halpern and Teague proposed the notion of *Nash equilibrium which survives the iterated deletion of weakly dominated strategies* as a security standard. But this notion is a weak guarantee, Kol and Naor [9] pointed out that a strategy that survives iterated deletion of weakly dominated strategies (iterated admissibility) may be undesirable, and suggested the notion of *strict Nash equilibrium* which requires that a player suffers losses if he deviates. However, the works [7,9] achieved the equilibrium notions they proposed by assuming the existence of simultaneous channels. After that Fuchsbauer et al. [5] achieved *a computational version of strict Nash equilibrium* in the synchronous model, but their scheme cannot resist against the attack of malicious adversaries.

Communication Channel. To design rational secret sharing protocols, the works [7,6,10] relied on the strong communication primitives of *simultaneous broadcast channels* to ensure the symmetry in the times that players get information. However, simultaneous communication is hard to be achieved, because it requires that all players move simultaneously and no player can observe other players' messages before he moves. Thus, several works [12,5,15] have been devoted to realizing rational secret sharing in the *synchronous* model, where the protocol runs in a series of rounds and only one player moves in each round. By delaying the signal indicating whether a given iteration is valid or not, the works [5,15] constructed efficient schemes over synchronous channels. By considering the mixture of rational players and honest players, Ong et al. [12] realized fairness over standard broadcast channels. However, these protocols may be problematic when some players are corrupted by malicious adversaries.

The Adversary Model. To guarantee the same security properties as in usual cryptographic model, some works [10,2,11]proposed to study rational secret sharing resisting against adversaries, there are two kinds. Abraham et al. [1] first considered rational players with unexpected utilities who do not respond to incentives the way we expect as adversaries. Then Altabari et al.[2] improved the protocol of [8] to resist against rational adversaries. Since malicious players in real-life executions behave arbitrarily and do not aim to get the secret, Lysyanskaya and Triandopoulos [10] proposed that it is appropriate to treat these players as malicious adversaries who behave irrationally and cannot be motivated. They realized multiparty computation for t-NCC functions in the mixed-behavior model consisting of adversaries and rational players, which can tolerate an adversary controlling at most $t-2$ players, but the solution is not suitable for rational secret sharing. Different from previous works, [11] modeled rational secret sharing in the repeated game (the game is repeated several times and players get a payoff for each game), and introduced a *repeated rational secret sharing* protocol tolerating malicious players. However, it is undesirable that the protocols in [10,1,2] rely on simultaneous communication, moreover, another noticeable point of all protocols above is that they can only achieve a Nash equilibrium surviving iterated deletion, which is too weak.

Problem Statement. Achieving a stronger equilibrium concept than Nash equilibrium, avoiding simultaneous communication and resisting against adversaries are three separate development directions of rational secret sharing. Current schemes have been developed tolerating adversaries, but they are less satisfactory in other two aspects. There is no improvement to rational secret sharing from all the three aspects.

1.2 Our Contribution

In this paper we design a t-out-of-n ($t \leq \lceil \frac{n}{2} \rceil$) rational secret sharing protocol tolerating an irrational adversary controlling a minority of players based on a publicly verifiable secret sharing (PVSS) scheme. It runs over synchronous (non-simultaneous) broadcast channels and achieves *a computational strict Nash equilibrium with respect to adversary structure $\mathcal{A}$*. Our work has advantages over previous solutions in the adversary model, we get rid of the simultaneous communication, moreover, the equilibrium concept we achieve is much stronger than the notion of Nash equilibrium that current works achieved.

2 Definition

2.1 Secret Sharing

In Shamir's t-out-of-n threshold secret sharing protocol, to share a secret $s \in \mathbb{Z}_q$, the dealer chooses a polynomial $f(x)$ of degree $t-1$, such that the constant term is s, then he publishes n distinct points $x_1, \ldots, x_n$ in field $\mathbb{Z}_q^*$ and sends $f(x_i)$ to P_i as his share. At least t players can recover the secret by using Lagrange interpolation, but less than t players cannot get any information about the secret. If the dealer is honest, the shares received by each player should be consistent. Here we describe what it means to be consistent.

Definition 1. *Given $S = \{s_1, \ldots, s_n\}$, $x_1, \ldots, x_n$ are n fixed distinct points, if there exists a polynomial $f(x)$ of degree $t-1$ ($t \leq n$), such that $f(x_i) = s_i, \forall s_i \in S$, then we say the that values in $\{s_1, \ldots, s_n\}$ are consistent, and that $f(0)$ is recovered from S consistently.*

2.2 Game Theoretic Model

We introduce a new game-theoretic model of *the mixed-behavior game with respect to adversary structure* for rational secret sharing where adversaries exist. Compared with the standard game model where all players are rational, our mixed-behavior game gives a description of the behaviors of adversaries.

Definition 2. *The mixed-behavior game Γ with respect to adversary structure* $\mathcal{A}$ consists of

- A finite set $P = \{P_i | i \in \{1, \ldots, n\}\}$ (the set of **players**), adversary structure $\mathcal{A} \subset 2^P$ satisfies that $\forall A \in \mathcal{A}$, if $A' \subseteq A$ then $A' \in \mathcal{A}$. For any adversary $A \in \mathcal{A}$ we denote rational player set $R = P \backslash A$.
- For each $P_i \in P$ a nonempty set AC_i (the set of **actions** available to P_i), and let $AC = AC_1 \times AC_2 \times \ldots \times AC_n$ be the set of action profiles.
- A set of sequences $H = \{(\boldsymbol{a}^k)_{k=0,\ldots,T} | \boldsymbol{a}^k \in AC, T \in \mathbb{N}\}$ that satisfies the following properties.
  - $\boldsymbol{a}^0 = \varnothing \in H$
  - If $(\boldsymbol{a}^k)_{k=0,\ldots,K} \in H$ and $L < K$ then $(\boldsymbol{a}^k)_{k=0,\ldots,L} \in H$
  - If an infinite sequence $(\boldsymbol{a}^k)_{k=0,\ldots}$ satisfies $(\boldsymbol{a}^k)_{k=0,\ldots,L} \in H$ for every positive integer L then $(\boldsymbol{a}^k)_{k=0,\ldots} \in H$

  Each member of H is a history. A history $(\boldsymbol{a}^k)_{k=0,\ldots,K} \in H$ is **terminal** if there is no $\boldsymbol{a}^{K+1}$ such that $(\boldsymbol{a}^k)_{k=0,\ldots,K+1} \in H$ or if it is infinite. The set of terminal histories is denoted Z.
- A function u_i (the **utility function** of player P_i) for each player $P_i \in P$, which assigns to each terminal history a real value, $u_i : Z \longrightarrow \mathbb{R}$. Let $\boldsymbol{u} = u_1 \times u_2 \times \ldots \times u_n$ be the utility profile.

We assume that in the game there exists a malicious adversary who takes no care of his outcome and behaves maliciously (does not behave according to what specified by the protocol). We do not limit his capability, he behaves arbitrarily and may deviate from the prescribed strategy even if doing so is not favorable to him. The adversary corrupts a subset of rational players before the game starts, gets all their information, takes full control of them and decides how to move in the following protocol. We use adversary structure $\mathcal{A}$ to model the subsets of players which are corrupted by the adversary, and treat all corrupted players as malicious throughout the protocol. Players do not know which subset of players has been corrupted and what the adversaries will do.

The game proceeds in a sequence of actions (AC). Players are perfectly informed of history H, which records the actions that have occurred, and then decide their plans of actions based on other players' behaviors. The action that P_i chooses in each step is determined by his strategy, it is a function from the non-terminal histories to his actions, $\sigma_i : H \backslash Z \longrightarrow AC_i$. S_i denotes the set of strategies of player P_i, let $S = S_1 \times \ldots \times S_n$ be the set of strategy profiles. A history after which no more choices have to be made is terminal, which corresponds to an outcome of the protocol. A utility function u_i is used to describe the preference of P_i over the outcomes, $u_i(h)$ refers to P_i's utility after the terminal history $h \in Z$. Since each terminal history is determined by a strategy profile, we define the *outcome* $O(\boldsymbol{\sigma})$ to be the terminal history that results when each player $P_i \in P$ follows the precepts of σ_i. That is, $O(\boldsymbol{\sigma})$ is the (possibly infinite) history $(\boldsymbol{a}^0, \ldots, \boldsymbol{a}^K) \in Z$ such that for $0 \leq k < K$, $\boldsymbol{\sigma}(\boldsymbol{a}^0, \ldots, \boldsymbol{a}^k) = \boldsymbol{a}^{k+1}$. Now we can define the utility function on strategies, let $u_i(\boldsymbol{\sigma}) = u_i(O(\boldsymbol{\sigma}))$.

2.3 Utility Assumption

The adversaries are irrational, unlike rational players, they have no preference over the outcomes, so they cannot be motivated and may follow the strategy which seems bad. They behave arbitrarily, unpredictably or even maliciously, and they do not aim to learn the secret, so we do not have to consider their payoffs and we treat their utilities as empty ones. We assume that rational players do not take care of the outputs of adversaries, and the utility of a rational player depends on both his own output and other rational players' outputs. Following [7], we assume that players prefer to get the secret first, and then prefer the fewest number of other rational players who get the secret, which can be formalized as follows. In the rational secret sharing protocol, for a given adversary $A \in \mathcal{A}$, a strategy profile $\boldsymbol{\sigma}$, $out(\boldsymbol{\sigma}) = (o_1, \ldots, o_n)$ such that (1) $o_i = 1$ *iff* player $P_i \in R$ can get the secret when all players stick to $\boldsymbol{\sigma}$, and $o_i = 0$ otherwise, (2) $o_j = 0$ if $P_j \in A$. For $P_i \in R$ it holds that:

1. $u_i(\boldsymbol{\sigma}) = u_i(\boldsymbol{\tau})$ if $out(\boldsymbol{\sigma}) = out(\boldsymbol{\tau})$
2. $u_i(\boldsymbol{\sigma}) > u_i(\boldsymbol{\tau})$ if $out_i(\boldsymbol{\sigma}) = 1$ and $out_i(\boldsymbol{\tau}) = 0$
3. $u_i(\boldsymbol{\sigma}) > u_i(\boldsymbol{\tau})$ if $out_i(\boldsymbol{\sigma}) = out_i(\boldsymbol{\tau})$, $out_j(\boldsymbol{\sigma}) \leq out_j(\boldsymbol{\tau})$ for all $j \neq i$ and there exists a $P_k \in R$, $k \neq i$ such that $out_k(\boldsymbol{\sigma}) < out_k(\boldsymbol{\tau})$

U_i^+ denotes the utility of P_i when P_i learns the secret but other rational players do not, U_i denotes the utility of P_i when all rational players learn the secret, U_i^- denotes the utility of P_i when P_i himself learns no secret. It follows that $U_i^+ > U_i > U_i^-$. Furthermore, we assume that the secret is chosen from domain $\mathbb{Z}_q$, players can guess the secret with probability $1/q$. U_i^r denotes the utility that P_i gets when he tries to guess the secret, $U_i^r = \frac{1}{q}U_i^+ + (1 - \frac{1}{q})U_i^-$. We assume that $U_i^r < U_i$, or else players may gain without running the protocol. Moreover, we assume there is a non-negligible difference between U_i and U_i^r. That is, there exists a polynomial $p(\cdot)$ such that for all sufficiently large k's it holds that $U_i > U_i^r + \frac{1}{p(k)}$.

2.4 Definition of Game-Theoretic Equilibrium

Definition 3. *We say that a function ν is negligible if for every constant $c \geq 0$ there exists an integer k_c such that $v(k) < k^{-c}$ for all $k \geq k_c$.*

Following [5], we define what it means to follow the protocol in our model first. Since players have bounded computing power, their strategies can be seen as probabilistic polynomial-time interactive Turing machines. We measure equivalence of strategies according to their views. Given the prescribed strategy profile $\boldsymbol{\sigma} = \sigma_1 \times \ldots \times \sigma_n$ of protocol Π, let $\boldsymbol{\sigma}_{R\setminus i} = \times_{P_j \in R \setminus P_i} \sigma_j$ denote the strategy profile of rational players except P_i. We have the following definition.

Definition 4. *Given an adversary $A \in \mathcal{A}$, denote $P_{R\setminus i} = \{P_j | j \neq i, P_j \in R\}$. Let the adversary A follow the strategy profile $\boldsymbol{\tau}_A \in S_A$, the rational players R follow the strategy profile $\boldsymbol{\sigma}_R$. Define the random variable $\mathsf{View}_{-i}^{\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A}$ as follows*:

> *Let* Trans *denote the messages sent by P_i not including any message sent by P_i after he writes to his output tape.* $\mathsf{View}_{-i}^{\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A}$ *includes the information given by the dealer to P_{-i}, the random coins of P_{-i} and the (partial) transcript* Trans.

Fix a strategy ρ_i and an algorithm T. Define the random variable $\mathsf{View}_{-i}^{T, \rho_i}$ as follows:

> *When the players interact, P_i follows ρ_i, $P_{R\setminus i}$ follow $\boldsymbol{\sigma}_{R\setminus i}$, adversaries follow $\boldsymbol{\tau}_A$. Let* Trans *denote the messages sent by P_i. Algorithm T, given the entire view of P_i, output an arbitrary truncation* Trans' of Trans *(defining a cut-off point and deleting any messages sent after that point).* $\mathsf{View}_{-i}^{T, \rho_i}$ *includes the information given by the dealer to P_{-i}, the random coins of P_{-i} and the (partial) transcript* Trans'.

Strategy ρ_i yields equivalent player with respect to Π, denoted $\rho_i \approx \Pi$, if $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$ there exists a PPT algorithm T such that for all PPT distinguishers D:

$$|Prob[D(1^k, \mathsf{View}_{-i}^{T, \rho_i}) = 1]| - |Prob[D(1^k, \mathsf{View}_{-i}^{\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A}) = 1]| \leq negl(k)$$

where $negl(k)$ is a negligible function.

From the definition above, a strategy which yields equivalent player with respect to Π should tolerate the deviations of a certain number of adversaries, but may differ from the prescribed strategy when the player can be sure that one rational player deviates, it may even deviate from σ after the player gets the output, and we call it *equivalent strategy* for short.

Intuitively, in the rational secret sharing protocol running over synchronous channels, the players who broadcast shares later have the privilege to identify the real secret. Once they learn the significant information, they are no longer afraid of being punished, and they would like to deviate from the protocol because of rationality. This deviation cannot be avoided, but it may cause a problem that not all the players can learn the secret in a protocol inducing a standard (strict) equilibrium in the synchronous model. The standard equilibrium can only guarantee that a rational player cannot prevent any player from learning the secret on condition that all other rational players follow the protocol. However, in the synchronous model the requirement that all remaining rational players follow cannot be satisfied, because they would like to deviate after they learn the secret. When a certain number of players deviate from the protocol, some players may not be able to learn the secret. Hence, the standard equilibrium is too weak for the synchronous model. To achieve fairness that all players can still get the secret in this situation, it should be required that even if some other rational players follow the equivalent strategies, deviating will not be better than following for any player, i.e. no one will be prevented from learning the secret.

In addition, since adversaries cannot be motivated to follow and their deviation may induce rational players to deviate, we enhance the standard notion of strict Nash equilibrium to tolerate adversaries. Considering computational limitations, we give the following definition of computational strict Nash equilibrium with respect to $\mathcal{A}$. It is appealing that rational players have an incentive not to deviate before outputting, and the fairness can be satisfied even if players deviate after getting the outputs. Let $Eqv_i(\boldsymbol{\sigma})$ be the union of σ_i and the set of the strategy ρ_i of P_i which yields equivalent with respect to $\mathit{\Pi}$, $Eqv_R(\boldsymbol{\sigma}) = \times_{P_i \in R} Eqv_i(\boldsymbol{\sigma})$.

Definition 5. *Let $\boldsymbol{\sigma}$ be the prescribed strategy profile of the protocol $\mathit{\Pi}$, $\mathit{\Pi}$ induces a computational strict Nash equilibrium with respect to the adversary structure $\mathcal{A}$ if it satisfies:*

1. *For each $P_i \in R$ and each deviating strategy $\sigma'_i \in S_i$ $\sigma'_i \not\approx \mathit{\Pi}$, it is satisfied that: $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$ there is a $c > 0$ such that $u_i(\sigma_i, \boldsymbol{\sigma}_{\boldsymbol{R \setminus i}}, \boldsymbol{\tau}_A) \geq u_i(\sigma'_i, \boldsymbol{\sigma}_{\boldsymbol{R \setminus i}}, \boldsymbol{\tau}_A) + \frac{1}{k^c}$ for infinitely many values k, that is, $u_i(\sigma_i, \boldsymbol{\sigma}_{\boldsymbol{R \setminus i}}, \boldsymbol{\tau}_A) - u_i(\sigma'_i, \boldsymbol{\sigma}_{\boldsymbol{R \setminus i}}, \boldsymbol{\tau}_A)$ is non-negligible.*
2. *For each $P_i \in R$, $\forall \rho_i \approx \mathit{\Pi}$ $(\rho_i \neq \sigma_i)$, it holds that $u_i(\rho_i, \boldsymbol{\rho}_{R \setminus i}, \boldsymbol{\tau}_A) \leq u_i(\sigma_i, \boldsymbol{\rho}_{R \setminus i}, \boldsymbol{\tau}_A) + negl(k)$, $\forall \boldsymbol{\rho}_{R \setminus i} \in Eqv_{R \setminus i}(\boldsymbol{\sigma})$, $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$.*

The sufficiently strong notion of computational strict Nash equilibrium with respect to $\mathcal{A}$ requires that no matter how the adversaries behave, a player suffers losses if he deviates before getting the output, and that when other rational players follow the equivalent strategies, he cannot increase his payoff by a non-negligible amount if he deviates after getting the output. The last point models the fact that we cannot force P_i to send correct messages once he can be sure that the protocol is finished. However, this definition guarantees that even if rational players deviate after outputting, the fairness can still be satisfied with probability $1 - \epsilon$ (ϵ is negligible).

3 Publicly Verifiable Secret Sharing

Consider such a problem in secret sharing: in many applications the dealer and the shareholders do not trust each other, the dealer may distribute incorrect shares in the distribution phase and shareholders may reveal incorrect shares in the reconstruction phase. These problems are also apparent in our mixed-behavior model, moreover the adversary may accuse when receiving consistent shares in order to halt the protocol. The publicly verifiable secret sharing (PVSS) scheme provides an approach for all players to verify that a share is consistent with other shares without revealing any information about the secret. Hence, it cannot only detect forged messages but also prevent the adversary from declaring an accusation against an honest dealer. We will use it as a tool for detecting deviations in our rational secret sharing protocol.

Let p and q denote large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q, and g, h, G, H denote the independently chosen generators of G_q such that no one knows $\log_g h, \log_G H$. A publicly known element $x_i \in \mathbb{Z}_q^*$ is assigned to each P_i. A commonly used method for verifying the consistency of shares in PVSS works as follows, the dealer shares c through $f(x) = c + a_1 x + \ldots + a_{t-1}x^{t-1}$, publishes the commitments to the coefficients $g^c, g^{a_1}, \ldots, g^{a_{t-1}}$, and sends the share $f(x_i)$ to P_i, it satisfies that $g^{f(x_i)} = g^c \prod_{j=1}^{t-1}(g^{a_j})^{x_i^j}$. However, this approach is problematic when being used in our RSS protocol, because whether c equals zero or not can be revealed from g^c, from which players can recognize the valid iteration in advance (some protocols set c to be zero in the valid iteration). Pedersen's VSS scheme requires the committer to compute $g^c h^r$ as a commitment to c by using r, denoted $E(c,r) = g^c h^r$. It is not suitable for our protocol either, because players can learn g^c before revealing their shares just after r is calculated.

In order to guarantee the privacy of the iteration status before the secret is recovered, g^c should be private. For this reason, we propose a publicly verifiable secret sharing scheme in this section, which satisfies that the shares can be verified, and most importantly, no information about g^c is revealed during the protocol under the Decision Diffie-Hellman assumption. Our construction for PVSS is based on Pedersen's VSS scheme [13] and Stadler's PVSS scheme [14], but we make some necessary modifications to it so that it is applicable for our RSS protocol. We use a generalization of the Chaum and Pedersen protocol [4] as a subprotocol to prove the equality of the contents of two commitments, denoted $DLEV(g, h, G, H, l_1, l_2)$. (Generators $g, h, G, H \in G_q$ are public.) The prover publishes two commitments, i.e. $l_1 = g^c h^r$ and $l_2 = G^c H^r$, then he proves that l_1, l_2 are both commitments to c for r as follows:

1. The prover chooses w, w' randomly, and sends $a_1 = g^w h^{w'}$ and $a_2 = G^w H^{w'}$ to the verifier.
2. The verifier chooses b randomly and sends it to the prover.

3. The prover opens $u = w - c \cdot b \ mod \ q$ and $u' = w' - r \cdot b \ mod \ q$.
4. The verifier checks that $a_1 = (g^u h^{u'}) \cdot l_1^b$ and $a_2 = (G^u H^{u'}) \cdot l_2^b$.

Lemma 1. *Two commitments $l_1 = g^{c_1} h^{r_1}$ and $l_2 = G^{c_2} H^{r_2}$ are published by the prover, under the Discrete Logarithm assumption, if the verifier accepts the verification $DLEV(g, h, G, H, l_1, l_2)$, then it means that $c_1 = c_2$ and $r_1 = r_2$. We say that the contents of l_1 and l_2 are equal.*

***Proof*.** We assume that the prover can find two two-tuples (c, r) and (c', r') such that $l_1 = g^c h^r$ and $l_2 = G^{c'} H^{r'}$ are accepted by the verifier. We can get that:

$$a_1 = g^w h^{w'} = g^u h^{u'} \cdot (g^c h^r)^b = g^{u+c \cdot b} \cdot h^{u'+r \cdot b}$$

satisfies $w = u + c \cdot b$, $w' = u' + r \cdot b$. Let $a_2 = G^w H^{w'}$, it also holds that:

$$a_2 = G^w H^{w'} = G^u H^{u'} \cdot (G^{c'} H^{r'})^b = G^w H^{w'} G^{(c'-c)b} H^{(r'-r)b}$$

and in particular $G^{(c'-c)b} H^{(r'-r)b} = 1$, then we can get that

$$\log_G H = \frac{c-c'}{r'-r} (mod \ q)$$

Therefore, if the prover can pass the verification when cheating, then computing discrete logarithm becomes feasible. □

Denote the initial protocol of Chaum and Pedersen [4] by $LEV(g, G, g^c, G^c)$, which is used to verify whether the discrete logarithm of g^c to the base g equals the discrete logarithm of G^c to the base G. It is similar to $DLEV(g, h, G, H, g^c h^r, G^c H^r)$ except $r = 0$ and is much more simple. We will use it for verification in our RSS protocol. We omit the description of it here which can be found in [4].

3.1 Verifiable Encryption

When the dealer distributes encrypted shares over a broadcast channel, he needs to make the encrypted shares publicly verifiable. We adopt the verifiable encryption scheme of [14]. It is a protocol for verifying that a pair (M, N) encrypts the discrete logarithm of a public element $V_i = G_i^{v_i}$, denoted $VES(v_i, G_i^{v_i})$. The details appear in Appendix A.

Under the Decision-Diffie-Hellman assumption, computing v_i from $G_i^{v_i}$ and (M, N) is hard. Moreover, the dealer can cheat successfully with negligible probability. We omit the proof here, which can be found in [14]. In fact, the negligible probability of a successful cheat has no bad influence on our rational secret sharing scheme, because P_i can verify the decrypted v_i by checking the commitment $G_i^{v_i}$, and then he opens his private key once it is forged, so that the malicious dealer is sure to be caught. Thus if the dealer can pass the verification we believe that the decryption is true.

3.2 PVSS Scheme

In this section, we give the construction of a t-out-of-n publicly verifiable secret sharing scheme. Under the Discrete Logarithm assumption, the dealer cannot succeed in sharing inconsistently. Given generators g, h of G_q which are chosen randomly so that no one knows $log_g h$, each P_i chooses $z_i \in_R \mathbb{Z}_q^*$ as his private key and publishes $G_i = g^{z_i}, H_i = h^{z_i}$ as his public keys.

Distribution of the shares

1. The dealer D wants to share a secret $c \in \mathbb{Z}_q$ among all players $P_1, \ldots, P_n$, and he chooses a $t-1$ degree polynomial $F(x)$ with coefficients in $\mathbb{Z}_q$:
$$F(x) = c + \alpha_1 x + \ldots + \alpha_{t-1} x^{t-1}$$
D computes $c_i = F(x_i)$ for $i \in \{1, \ldots, n\}$ and publishes a commitment to c for a randomly chosen value $r \in \mathbb{Z}_q$: $E_0 = E(c, r) = g^c h^r$.
2. D chooses $\beta_1, \ldots, \beta_{t-1} \in \mathbb{Z}_q$ randomly and broadcasts commitments $E_i = E(\alpha_i, \beta_i) = g^{\alpha_i} h^{\beta_i}$ to each α_i for $i \in \{1, \ldots, t-1\}$. D can get another $t-1$ degree polynomial $R(x) = r + \beta_1 x + \ldots + \beta_{t-1} x^{t-1}$.
3. D computes $r_i = R(x_i)$ and sends (c_i, r_i) to player P_i for $i \in \{1, \ldots, n\}$ by applying the verifiable encryption scheme, D calculates and sends out cipher-texts, then each P_i recovers c_i and r_i from them.
4. D also publishes commitments $A_i = G_i^{c_i}, B_i = H_i^{r_i}$ for $i \in \{1, \ldots, n\}$.

Verification of the shares

1. The verification of $G_i^{c_i} H_i^{r_i}$
D proves to all players that the contents of $g^{c_i} h^{r_i}$ and $A_i B_i = G_i^{c_i} H_i^{r_i}$ are equal through $DLEV(g, h, G_i, H_i, g^{c_i} h^{r_i}, A_i B_i)$ for $i \in \{1, \ldots, n\}$. In fact, players can compute $g^{c_i} h^{r_i}$ from the commitments:
$$g^{c_i} h^{r_i} = \prod_{j=0}^{t-1} E_j^{x_i^j}$$
2. The verification of c_i and r_i
D runs the protocol $VES(c_i, G_i^{c_i})$ and $VES(r_i, H_i^{r_i})$, if he can pass the verification, then P_i receives a share consistent with others.

Reconstruction

Each P_i broadcasts c_i and r_i, together with a proof that the shares published are the discrete logarithm of A_i to G_i and the discrete logarithm of B_i to H_i respectively. After all shares have been verified, players reconstruct c by using Lagrange interpolation, $c = \Sigma_{i=1}^{t} \gamma_i c_i$. ($\gamma_i$ is a Lagrange coefficient)

In the distribution phase, the dealer broadcasts the commitments to the coefficients $g^{\alpha_i}h^{\beta_i}$, from which the consistency of shares can be verified, and distributes the shares c_i and r_i by using the verifiable encryption scheme. In the verification phase, the dealer proves that the decrypted c_i, r_i are correct, and that the shares are distributed consistently through the polynomials that he commits to. Under the assumption of Decision Diffie-Hellman, no information about the secret is revealed from the commitments. Moreover, similar to [13], if the dealer passes the verification then the consistency of shares can be guaranteed. A proof of the following theorem appears in Appendix B.

Theorem 1. *Under the Decision Diffie-Hellman assumption, the PVSS scheme is secure: (1) If the dealer passes the verification, any subset of at least t players can reconstruct the same secret. (2) Any subset of less than t players cannot get any information about the secret.*

4 Rational Secret Sharing Protocol

In this section we give a t-out-of-n ($t \leq \lceil \frac{n}{2} \rceil$) rational secret sharing protocol that is resilient to an adversary corrupting less than $t-1$ players. We only need the existence of synchronous broadcast channels (but non-simultaneous), and assume all players to be computationally bounded. After the initialization, our protocol runs in a sequence of iterations, which is a frequently used technique for RSS, with the property that the secret s can be recovered in the valid iteration, and no information about s is revealed in the invalid iteration.

Our scheme depends on the masking of the secret in each iteration. Following the same high-level approach as in [3,15], players recover a ″one-time″ secret $s + c$ in each iteration, where c is negotiated by a part of players randomly and is unknown to them, and players can get s only when $c = 0$. To run the protocol over a synchronous channel, we require players to identify whether the current iteration is valid or not after reconstructing $s + c$. The key in this process is that no information about c except that whether c equals 0 or not should be revealed when players check c. Players do so by verifying whether $g^{c\kappa}$ equals 1 or not (κ is a non-zero value) so as to keep s private. If $g^{c\kappa} = 1$, players can be convinced that the reconstructed $s + c$ equals s, otherwise the given iteration is invalid and players cannot learn c from $g^{c\kappa}$.

Different from the previous protocol where there is no adversary, our goal is to motivate all rational players to follow even if adversaries try to induce them to deviate. To that end, we require the above PVSS scheme, through which players verify the correctness of shares and catch a minority of deviations, and we punish deviating players by disqualifying them. These methods guarantee that a player decreases his payoff by deviating independently of the behaviors of adversaries.

4.1 Construction

Let g,h be the independently chosen generators of G_q, hence no player knows the discrete logarithm of h to g. Each P_i chooses $z_i \in_R \mathbb{Z}_q^*$ as his private key and publishes $G_i = g^{z_i}$ and $H_i = h^{z_i}$ as his public keys.
Initialization. The dealer only needs to be active in the initial stage. To share $s \in \mathbb{Z}_q$, the dealer chooses a $t-1$ degree polynomial $f(x) = s + a_1 x + \ldots + a_{t-1}x^{t-1}$ with coefficients in $\mathbb{Z}_q$ randomly. The dealer calculates $s_i = f(x_i)$ and sends it to P_i as his share, and publishes a commitment to each s_i: $X_i = G_i^{s_i}$.

Each Iteration. Each iteration includes two phases: new shares generation phase and reconstruction phase. In the first phase players negotiate about a random value c and generate the shares of the ″one-time″ secret $s + c$ for this iteration. In the second phase the unknown value $s+c$ is reconstructed first. After verifying the validity of shares, players mask c and g^c by randomly choosing $\kappa_i \in \mathbb{Z}_q^*$ $(i \in \{1, \ldots, n\})$ and publishing $g^{c\kappa_1 \cdots \kappa_n}$, which equals 1 in the valid iteration. We give the formal specification of the protocol in the l-th iteration as follows:

New shares generation

1. Players choose a subset of t players randomly together such that each P_i is chosen with probability $\frac{t}{n}$, w.l.o.g denoted $P' = \{P_1, \ldots, P_t\}$. Each $P_i \in P'$ chooses $\widetilde{c}_i \in \mathbb{Z}_q$ randomly, such that $\widetilde{c}_i$ equals 0 with probability δ_0 and equals each $d \in \mathbb{Z}_q^*$ with probability $\frac{1-\delta_0}{q-1}$, and then shares $\widetilde{c}_i$ through the PVSS scheme described above.

   In the PVSS scheme, P_i commits to $\widetilde{c}_i$ for a randomly chosen $\widetilde{r}_i \in \mathbb{Z}_q$. Denote by $\widetilde{C}_i(x) = \widetilde{c}_i + \widetilde{\alpha}_{i1}x + \ldots + \widetilde{\alpha}_{i(t-1)}x^{t-1}$ the polynomial for sharing $\widetilde{c}_i$. P_i chooses $\widetilde{\beta}_{ij} \in \mathbb{Z}_q$ and gets a polynomial $\widetilde{R}_i(x) = \widetilde{r}_i + \widetilde{\beta}_{i1}x + \ldots + \widetilde{\beta}_{i(t-1)}x^{t-1}$. P_i sends $\widetilde{c}_{ih} = \widetilde{C}_i(x_h)$ and $\widetilde{r}_{ih} = \widetilde{R}_i(x_h)$ to P_h, and publishes the following commitments to the shares $(j \in \{1, \ldots, t-1\}, h \in \{1, \ldots, n\})$: $\widetilde{E}_{i0} = E(\widetilde{c}_i, \widetilde{r}_i)$, $\widetilde{E}_{ij} = E(\widetilde{\alpha}_{ij}, \widetilde{\beta}_{ij})$, $G_h^{\widetilde{c}_{ih}}$, $H_h^{\widetilde{r}_{ih}}$.

   Disqualify P_i if he fails. Halt the protocol and output a random guess of the secret if there are at most $n-t+1$ players stay.
2. Calculate the shares of $c = \sum_{k=1}^{t} \widetilde{c}_k$ and $r = \sum_{k=1}^{t} \widetilde{r}_k$ by adding corresponding shares as follows:
$$c_i = \textstyle\sum_{k=1}^{t} \widetilde{c}_{ki}, \ r_i = \sum_{k=1}^{t} \widetilde{r}_{ki}$$
3. Calculate the commitments to the shares of c and r as follows $(i \in \{1, \ldots, n\}$,$j \in \{0, \ldots, t-1\})$:
$$E_j = \textstyle\prod_{k=1}^{t} \widetilde{E}_{kj}, \quad Y_i = G_i^{c_i} = \prod_{k=1}^{t} G_i^{\widetilde{c}_{ki}}$$
4. Each P_i calculates his share of $s + c$ by adding s_i and c_i.

Reconstruction

Stage 1: *reconstructing* $s^{(l)} = s + c$

Players take turns to broadcast their shares of $s + c$, and verify the authenticity of P_i's share by checking: $G_i^{s_i+c_i} = X_i Y_i$. It is required that:

(1) Disqualify P_i if his share cannot satisfy the equation above, or else we say that $s_i + c_i$ is credible.

(2) Halt the protocol if the credible shares broadcasted are inconsistent or at most $n - t + 1$ credible shares have been broadcasted, and then output a random guess of the secret.

(3) Reconstruct $s^{(l)}$ from these credible shares consistently if there are more than $n - t + 1$ credible shares which are consistent.

Stage 2: *checking* c

1. Each P_i chooses $\kappa_i \in_R \mathbb{Z}_q^*$ and publishes $K_i = g^{\kappa_i}$. Let $\kappa = \prod_{i=1}^n \kappa_i$.
2. Players compute $g^{c_i\kappa}$ for $i \in \{1, \ldots, n\}$ as follows. (Take $g^{c_i\kappa}$ as an example)
   (a) P_i calculates $g^{c_i\kappa_i}$ and publishes it, then P_i verifies its correctness through $LEV(G_i, K_i, G_i^{c_i}, g^{c_i\kappa_i})$, that is he verifies the discrete logarithm of $G_i^{c_i}$ to the base G_i is equivalent to the discrete logarithm of $g^{c_i\kappa_i}$ to the base g^{κ_i}.
   (b) Players take turns to calculate as follows, start from P_{i+1}, then P_{i+2},...,P_n,P_1,...,P_{i-1}. After P_{i-1} finishes, they can get $g^{c_i\kappa}$. We assume that it is P_j's turn now. P_j calculates and publishes $g^{c_i\kappa_i\cdots\kappa_j}$ after $g^{c_i\kappa_i\cdots\kappa_{j-1}}$ has been published by P_{j-1}, then P_j implements $LEV(g, g^{c_i\kappa_i\cdots\kappa_{j-1}}, K_j, g^{c_i\kappa_i\cdots\kappa_j})$.
   (c) We say $g^{c_i\kappa}$ is credible if it is calculated successfully. If someone fails when computing $g^{c_i\kappa}$, then players disqualify him and restart Stage 2 unless $g^{c\kappa}$ has been calculated, in which case players enter Step 3.
   (d) Halt the protocol and output $s^{(l)}$ if at most $n - t + 1$ players stay or the credible shares that have been calculated are inconsistent.
3. Reconstruct $g^{c\kappa}$ from credible shares (w.l.o.g from the first t shares) as follows: $g^{c\kappa} = \prod_{i=1}^t (g^{c_i\kappa})^{\gamma_i}$, γ_i is a Lagrange coefficient.
4. Output $s^{(l)}$ and terminate the protocol if $g^{c\kappa}$ equals 1, otherwise, proceed to the next iteration after disqualifying the deviating players.

Remark 1. *The fact that the parameter c is negotiated by t players instead of all players is due to malicious behavior of adversaries. An adverse P_i can make the probability of $c = 0$ very low by not choosing $\widetilde{c}_i$ according to δ_0. However, when these t players are all rational players who follow the protocol, $c = \sum_{i=1}^t \widetilde{c}_i$ equals 0 with the probability $p = \frac{1}{q} + \frac{(\delta_0 \cdot q - 1)^t}{q(q-1)^{t-1}}$ (c may equal zero when each $\widetilde{c}_i$ equals zero or one of the $\widetilde{c}_i$ equals the opposite of the sum of the others). According to our protocol, c is negotiated by t rational players with the probability at least $\binom{n-t+2}{t}/\binom{n}{t}$. Since the protocol terminates when $c = 0$, we can avoid very low probability of termination by setting δ_0 appropriately.*

Remark 2. *The use of $g^{c\kappa}$ for checking c is to ensure the privacy of c in the invalid iteration. Obviously, broadcasting the shares of c directly will not do. In addition, it is more secure to open $g^{c\kappa}$ than open g^c, even if players can learn $c\kappa$ from some special $g^{c\kappa}$, they cannot learn c from $c\kappa$ because of the unknown κ.*

4.2 Analysis

It is obvious that the deviations of adversaries will be caught if all rational players follow the protocol. Since the adversaries are at most $t-2$, the correct shares revealed are more than t, so that an authorized subset of at least t honest players can recover the real secret from their shares at last. Now we first prove the privacy of the protocol, i.e. during the protocol an unauthorized subset cannot learn any information about the secret.

Theorem 2. *Under the Discrete Logarithm assumption, no information about s is revealed to any subset of less than t players before the reconstruction stage of the last iteration. Under the Decision Diffie-Hellman assumption, no information about c is revealed to any subset of less than t players before the second stage of the reconstruction phase.*

Proof. In the first phase, the commitments $G_i^{s_i}, G_i^{c_i}, H_i^{r_i}, g^{c_i}h^{r_i}$ are published. However, players cannot compute s_i, c_i from these commitments, or else computing discrete logarithm becomes feasible. Thus less than t players have no enough shares and cannot reconstruct s or c. In the first stage of reconstruction phase, $s+c$ can be reconstructed, but players can only learn the sum of unknown s_i and c_i, so s and c keeps private. When checking c in the invalid iteration, only $g^{c\kappa} \neq 1$ can be calculated, under the discrete logarithm assumption, no information about c except that c does not equal 0 is revealed, thus s cannot be calculated from $s+c$.

In the PVSS scheme, no information about g^{c_i} is revealed under the Decision Diffie-Hellman assumption, so even if $c=0$ players cannot learn it. Moreover, c and g^c keep private when players reconstruct $s+c$. Thus players cannot learn c in advance. The result holds. □

From the analysis above, it is obvious that no information about the secret is reveal in the invalid iteration because s is masked by using $s+c$, and no one can learn the iteration status in advance. Next we need to prove that the protocol leads to a computational strict Nash equilibrium. We first consider a problem that in our mixed-behavior model whether the rational secret sharing protocol can resist against $t-1$ malicious players or not. As pointed out by [10], if the adversary can control $t-1$ players, he may send $t-1$ shares to $n-2t+2$ rational players, so that these rational players can recover the secret without running the protocol after verifying the validity of the shares by themselves. This situation is extremely undesirable, the remaining $t-1$ players cannot recover the secret any longer because rational players would not like to participant in the protocol after learning the secret. To avoid this problem we allow the adversary to control at most $t-2$ players. Now we give the proof of our result. (We assume that a given iteration is valid with probability β.)

Theorem 3. *The (t, n) RSS protocol Π runs in the mixed-behavior model where rational players and irrational adversaries coexist, $\mathcal{A}$ denotes the adversary structure which consists of subsets of less than $t-1$ players, Π induces a computational strict Nash equilibrium with respect to $\mathcal{A}$, if there exists a polynomial p such that for all sufficiently large k's it holds that $\frac{U_i-U_i^r}{U_i^+-U_i^r} - \delta_0 > \frac{1}{p(k)}$.*

Proof. In RSS, it is unavoidable that cryptographic primitives may be broken before the last iteration with negligible probability, because RSS may execute in an exponential number of iterations. Thus, we consider the case that the cheating players share parameters inconsistently without being caught. Let $\varepsilon(k)$ be the negligible probability that rational P_i succeeds in doing so. Let $A \in \mathcal{A}$ be any adversary set of size at most $t-2$, and $\varepsilon'(k)$ denotes the negligible probability that at least one of the adversaries succeeds in deviating. If adversaries share inconsistently then rational players cannot recover the secret no matter whether they follow, because they cannot distinguish incorrect shares. Otherwise, as long as all rational players follow the protocol, all deviations of $(\leq t-2)$ adversaries are sure to be caught and the secret can be recovered. Thus, when all rational players follow, the utility that P_i gets is at least $U_i^r \varepsilon'(k) + U_i(1-\varepsilon'(k))$.

$\boldsymbol{\sigma}$ denotes the prescribed strategy profile in the protocol Π. $\boldsymbol{\tau}_A$ denotes an arbitrary strategy of a given adversary $A \in \mathcal{A}$. We assume that rational players except P_i stick to the prescribed protocol $\boldsymbol{\sigma}$, P_i follows the deviating strategy $\sigma_i' \not\approx \Pi$, which denotes the strategy that P_i deviates from σ_i before outputting the real secret. We first prove that σ_i is strictly better than σ_i' no matter how adversaries behave. We need to consider the following three deviating cases: (1) P_i shares inconsistently through the PVSS scheme or keeps silent when players generate new shares. (2) P_i broadcasts a forged share of $s+c$ or keeps silent in the reconstruction stage. (3) P_i deviates when checking c.

We analyze the situation where P_i deviates in the first case. Caught_i denotes the fact that P_i is caught deviating. Valid denotes the fact that P_i deviates in the valid iteration. Before the reconstruction phase P_i will be caught once he keeps silent. We now analyze the situation where P_i tries to share inconsistently. If P_i succeeds in distributing inconsistent shares while adversaries do not, then the protocol will halt in the reconstruction stage. P_i will be the only player who learns the secret if the current iteration is valid with probability β, but he cannot learn the secret in the invalid iteration with probability $1-\beta$. If P_i cannot succeed in deviating, then he will be disqualified, which happens with probability $1-\varepsilon(k)$. The expected utility that P_i gets by deviating is

$$\begin{aligned}
& u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) \\
\leq\ & U_i^r \cdot \mathsf{Prob}[\overline{\mathsf{Caught}_A}] + U_i^+ \cdot \mathsf{Prob}[\overline{\mathsf{Caught}_i} \wedge \mathsf{Caught}_A \wedge \mathsf{Valid}] \\
& + U_i^r \cdot \mathsf{Prob}[\overline{\mathsf{Caught}_i} \wedge \mathsf{Caught}_A \wedge \mathsf{Invalid}] \\
& + U_i^r \cdot (\mathsf{Prob}[\mathsf{Caught}_i \wedge \mathsf{Caught}_A \wedge \mathsf{Valid}] \\
& + \mathsf{Prob}[\mathsf{Caught}_i \wedge \mathsf{Caught}_A \wedge \mathsf{Invalid}])
\end{aligned}$$

$$\begin{aligned}&= U_i^r\varepsilon'(k) + U_i^+\varepsilon(k)(1-\varepsilon'(k))\beta + U_i^r\varepsilon(k)(1-\varepsilon'(k))(1-\beta)\\ &\quad +U_i^r[(1-\varepsilon(k))(1-\varepsilon'(k))\beta + (1-\varepsilon(k))(1-\varepsilon'(k))(1-\beta)]\\ &= U_i^r\varepsilon'(k) + U_i(1-\varepsilon'(k)) + (U_i^r - U_i)(1-\varepsilon'(k))\\ &\quad +(U_i^+ - U_i^r)\varepsilon(k)(1-\varepsilon'(k))\beta\end{aligned}$$

where $(U_i^+ - U_i^r)\varepsilon(k)(1-\varepsilon'(k))\beta = \eta(k)$ is negligible.

It follows that $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$,

$$u_i(\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A) \geq u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) + (U_i - U_i^r)(1-\varepsilon'(k)) - \eta(k)$$

We can notice that $(U_i - U_i^r)(1-\varepsilon'(k))$ is positive and non-negligible, $\eta(k)$ is negligible, so that $(U_i - U_i^r)(1-\varepsilon'(k)) - \eta(k)$ is positive and non-negligible. Thus, following the protocol is strictly better than deviating for P_i.

In the second case, (1) We assume that the shares are generated consistently, P_i forges his share or does not broadcast anything in stage 1, but he will be caught and be disqualified. Even if all $t-2$ adversaries deviate when P_i deviates, there are at least t players who follow the protocol, so the deviation of P_i will not interface with other players' reconstruction of $s+c$. Thus, if $s+c$ happens to be the real secret, P_i will get utility U_i, if it is fake then P_i can only guess the secret and get utility U_i^r. Considering the probability that adversaries share inconsistently in the first phase, the expected utility of P_i with this deviation is at most $U_i^r\varepsilon'(k) + (1-\varepsilon'(k))(\beta U_i + (1-\beta)U_i^r)$, we have $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$,

$$u_i(\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A) \geq u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) + (1-\varepsilon'(k))(1-\beta)(U_i - U_i^r)$$

(2) After $s+c$ has been recovered, P_i can deviate by quitting and outputting $s+c$. However, all players have gotten $s+c$, they can output the real secret in the valid iteration and the fake secret in the invalid iteration. In this case, P_i can get $U_i\beta + U_i^r(1-\beta)$, we have $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$,

$$u_i(\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A) \geq u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) + (1-\varepsilon'(k)-\beta)(U_i - U_i^r)$$

P_i can also output $s + c - \widetilde{c}_i$ when c is negotiated by P_i and other $t-1$ players. If $c - \widetilde{c}_i = 0$, then only P_i can learn the secret, and the maximum probability is δ_0 (when $t-2$ of these $t-1$ players are adversaries who always choose 0). Thus, the utility P_i can get is at most $U_i^+\delta_0 + U_i^r(1-\delta_0)$, we have $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$,

$$u_i(\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A) \geq u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) + \varepsilon'(k)(U_i^r - U_i) + \tfrac{1}{p(k)}(U_i^+ - U_i^r)$$

Since $(1-\varepsilon'(k))(1-\beta)(U_i - U_i^r)$, $(1-\varepsilon'(k)-\beta)(U_i - U_i^r)$, $\varepsilon'(k)(U_i^r - U_i) + \frac{1}{p(k)}(U_i^+ - U_i^r)$ are positive and non-negligible, following the protocol is strictly better than deviating.

In the third case, P_i deviates when checking c. There are two possible cases. The first one is that P_i computes $g^{c_j\kappa}$ $(j \in \{1,\ldots,n\})$ with a value different from what he commits to. The second one is that P_i keeps silent. In both cases, P_i will be caught and be disqualified, so we do not distinguish between them. If P_i deviates before t credible shares $g^{c_j\kappa}$ have been calculated, then players restart Stage 2 at once after disqualifying P_i. In this situation, even if $t-2$ adversaries deviate, the protocol will halt and all players will have the same output value. Thus P_i gets U_i in the valid iteration, but can only get U_i^r in the invalid iteration, and his deviation results in the expected utility $U_i\beta + U_i^r(1-\beta)$. If P_i follows the protocol then $g^{c\kappa}$ can be calculated. If the current iteration is valid then all players learn the secret, or else they continue the protocol that may be interrupted when adversaries share inconsistently in the subsequent iterations with negligible probability. Thus P_i gets $U_i\beta + (1-\beta)(\varepsilon'(k)U_i^r + (1-\varepsilon'(k))U_i)$ by following. We can get that $\forall A \in \mathcal{A}$, $\forall \boldsymbol{\tau}_A \in S_A$,

$$u_i(\boldsymbol{\sigma}_R, \boldsymbol{\tau}_A) \geq u_i(\sigma_i', \boldsymbol{\sigma}_{R\setminus i}, \boldsymbol{\tau}_A) + (1-\varepsilon'(k))(1-\beta)(U_i - U_i^r)$$

It follows that P_i suffers losses by deviating, since $(1-\varepsilon'(k))(1-\beta)(U_i - U_i^r)$ is positive and non-negligible.

As we analyzed above, P_i will decrease his payoff by a non-negligible amount if he deviates before learning the secret no matter how adversaries behave. Thus, the deviating strategy σ_i' is strictly worse than the prescribed strategy σ_i.

We next consider the scenario where all rational players follow the strategy profile $\boldsymbol{\rho}_R \in Eqv_R(\boldsymbol{\sigma})$, that is all rational players would deviate from the protocol after they can output the secret, and they would not deviate before they can output the secret when there are less than $t-1$ players deviate. We show that in this situation each P_i cannot gain by deviating after learning the secret. When checking c in the valid iteration, players calculate the share $g^{c_i\kappa}$ one by one, w.l.o.g. we assume that the shares are calculated in this order $g^{c_1\kappa}, ..., g^{c_n\kappa}$. If the first $t-1$ shares have been calculated and published, then P_{t-1} will be the first player who learns the iteration status after finishing the computation of $g^{c_t\kappa}$. If the current iteration is valid, then P_{t-1} may deviate by keeping silent or forging shares. However his lie is sure to be caught. If $t-2$ players (may be adversaries) have deviated, then the protocol halts and all players output $s^{(r)}$. If less than $t-2$ players have deviated, then players restart stage 2. Similarly, in the resumption of stage 2 the player who finishes the computation of the t'th share can learn the secret in advance and then would like to deviate. No matter what the adversaries do and no matter whether P_i deviates or not, at last the following two scenarios occur: (1) The protocol halts when the $(t-1)$'th deviating player occurs. In this situation all players can output the real secret, because the secret has been reconstructed during the first stage of reconstruction phase. (2) Less than $t-1$ players deviate, at least $n-t+1$ credible shares are calculated and $g^{c\kappa}$ is reconstructed. In both scenarios above, all players can learn the secret at last. If the current iteration is invalid, $\boldsymbol{\rho}$ is just $\boldsymbol{\sigma}$. In addition, P_i can get the secret s directly by breaking the cryptographic primitives with negligible probability. When some $t-2$ players keep silent, P_i can increase his utility from U_i to U_i^+ by quitting. However, since it happens with negligible probability, P_i can only increase his overall utility by a negligible amount by following ρ_i.

From the above analysis we can know that P_i cannot gain by deviating after learning the secret, even if all other rational players follow the equivalent strategies. For each rational player P_i, $\forall \boldsymbol{\rho}_R \in Eqv_R\boldsymbol{\sigma}$, it holds that:

$$u_i(\boldsymbol{\rho}_R, \boldsymbol{\tau}_A) \leq u_i(\sigma_i, \boldsymbol{\rho}_{R\setminus i}, \boldsymbol{\tau}_A) + negl(k),\ \forall A \in \mathcal{A},\ \forall \boldsymbol{\tau}_A \in S_A$$

The negligible term is derived from the broken of cryptographic primitives.

Therefore, we can conclude that the prescribed strategy of the protocol induces a computational strict Nash equilibrium with respect to adversary structure $\mathcal{A}$.□

4.3 Discussion

Equilibrium. Our scheme is the first to induce a computational strict Nash equilibrium with respect to adversary structure $\mathcal{A}$. This solution concept extends the notion of strict Nash equilibrium to resist against adversaries, it is much stronger than the notion of Nash equilibrium surviving iterated deletion of weakly dominated strategies, which previous protocols tolerating adversaries ([10,11,2]) can only achieve.

Communication Channel. Our RSS protocol only needs synchronous broadcast channels. However, previous works with respect to adversaries [10,2,1] required simultaneous communication.

Adversary Resilience. Our protocol motivates all rational players to follow the protocol independently of the attacks of adversaries. Compared with previous works, we do not limit the ability of adversaries, and our solution guarantees the same properties as in cryptography.

5 Conclusion

We show how to realize a t-out-of-n secret sharing protocol that is resilient to irrational adversaries over standard (synchronous) broadcast channels in this paper. We rely on the publicly verifiable secret sharing scheme to detect deviations, so that the reconstruction of the secret will not be interrupted even if a small number of players deviate. Moreover, the deviating players are punished by being disqualified. Compared with Nash equilibrium that current works with respect to adversaries can achieve, our protocol achieves an enhanced notion of computational strict Nash equilibrium with respect to adversary structure $\mathcal{A}$. In addition, we can tolerate less than one half of all players being corrupted by a malicious adversary without using simultaneous communication.

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: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006,, pp. 53–62. ACM, New York (2006)
2. Altabari, N., Krohmer, A., Molter, H., Tarrach, T.: A rational secret sharing scheme robust against malicious players (2009)
3. Asharov, G., Lindell, Y.: Utility dependence in correct and fair rational secret sharing. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 559–576. Springer, Heidelberg (2009)

4. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)
5. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient rational secret sharing in standard communication networks. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 419–436. Springer, Heidelberg (2010)
6. Dov Gordon, S., Katz, J.: Rational secret sharing, revisited. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 229–241. Springer, Heidelberg (2006)
7. Halpern, J.Y., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC 2004, pp. 623–632 (2004)
8. 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)
9. Kol, G., Naor, M.: Games for exchanging information. In: STOC, pp. 423–432 (2008)
10. Lysyanskaya, A., Triandopoulos, N.: Rationality and adversarial behavior in multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
11. Maleka, S., Shareef, A., Pandu Rangan, C.: Rational secret sharing with repeated games. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 334–346. Springer, Heidelberg (2008)
12. Ong, S.J., Parkes, D.C., Rosen, A., Vadhan, S.: Fairness with an honest minority and a rational majority. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 36–53. Springer, Heidelberg (2009)
13. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
14. Stadler, M.: Publicly Verifiable Secret Sharing. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 190–199. Springer, Heidelberg (1996)
15. Zhang, Y., Tartary, C., Wang, H.: An efficient rational secret sharing scheme based on the Chinese remainder theorem. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. LNCS, vol. 6812, pp. 259–275. Springer, Heidelberg (2011)
16. Zhang, Z., Liu, M.: Unconditionally secure rational secret sharing in standard communication networks. In: Rhee, K.-H., Nyang, D. (eds.) ICISC 2010. LNCS, vol. 6829, pp. 355–369. Springer, Heidelberg (2011)

Appendix A: Verifiable Encryption

It is a protocol for verifying that a pair (M, N) encrypts the discrete logarithm of a public element $V_i = G_i^{v_i}$, denoted $VES(v_i, G_i^{v_i})$.

Let q be a large prime so that $p' = (q-1)/2$ is also prime, G_i is a generator of G_q, f is a fixed element of order p' in $\mathbb{Z}_q^*$, P_i chooses $z_i' \in_R \mathbb{Z}_{p'}$ as his private key and publishes $y_i = f^{z_i'} (mod\ q)$ as his public key. To encrypt $v_i \in \mathbb{Z}_q^*$, the dealer D chooses $\varepsilon_i \in \mathbb{Z}_{p'}$ randomly and computes $(M, N) = (f^{\varepsilon_i}, v_i^{-1} \cdot y_i^{\varepsilon_i})$.

P_i can decrypt (M, N) by calculating

$$v_i = (f^{\varepsilon_i})^{z_i'} / (v_i^{-1} \cdot y_i^{\varepsilon_i})$$

Then the dealer proves to all players that the decryption of (M, N) is just the discrete logarithm of $V_i = G_i^{v_i}$ to G_i. Actually, if (M, N) equals $(f^{\varepsilon_i}, v_i^{-1} \cdot y_i^{\varepsilon_i})$, then it holds that

$$V_i^N = (G_i^{v_i})^{v_i^{-1} \cdot y_i^{\varepsilon_i}} = G_i^{y_i^{\varepsilon_i}}$$

So the dealer needs to prove that the double discrete logarithm of V_i^N to the base G_i, y_i equals the discrete logarithm of M to the base f. They repeat the following scheme T times:

1. The dealer randomly chooses $w \in \mathbb{Z}_{p'}$, publishes $t_f = f^w$ and $t_{G_i} = G_i^{y_i^w}$.
2. The verifiers send $b \in_R \{0, 1\}$ to the dealer.
3. The dealer responses with $r = w - b \cdot \varepsilon_i$.
4. The verifiers check
   (a) $t_f = f^r \cdot M^b$=f^w.
   (b) $t_{G_i} = G_i^{y_i^r}$ when $b = 0$
   (c) $t_{G_i} = (V_i^N)^{y_i^r}$ when $b = 1$

Appendix B: Proof of Theorem 1

We recall the Diffie-Hellman assumption and the Decision Diffie-Hellman assumption briefly. The Diffie-Hellman assumption states that given g^α, g^β it is infeasible to compute $g^{\alpha\beta}$. The Decision Diffie-Hellman assumption states that given g^α, g^β, it is infeasible to determine whether a given g^γ equals $g^{\alpha\beta}$ or not.

Firstly we consider the privacy of shares. In the distribution phase the dealer publishes commitments to each c_i for r_i, according to [13] the commitment $E(c_i, r_i)$ protects the privacy of c_i, r_i unconditionally. The dealer also publishes the commitments $G_i^{c_i}$ and $H_i^{r_i}$, c_i and r_i keep private under the Discrete Logarithm assumption. Furthermore, it follows from the soundness of Stadler's verifiable encryption scheme that no information about c_i and r_i is revealed from the encrypted values under the assumption of Discrete logarithm and Decision Diffie-Hellman. We can get that, assume that computing discrete logarithm is hard and breaking ElGamal cryptosystem is hard, players cannot compute other players' shares, so that less than t players cannot compute c. This can be expressed by the following theorem. Let $shares_B$ denote all shares received by players in B from the dealer during the protocol.

Lemma 2. *In the (t, n) public verifiable secret sharing scheme, for any player subset B of size less than t, it holds that*

$$Pr[players\ in\ B\ learn\ c | shares_B] = Pr[players\ in\ B\ learn\ c]$$

Proof. We consider the subset of size $t - 1$ first, let $B = \{P_1, \ldots, P_{t-1}\}$. Note that $shares_B = (c_1, r_1, \ldots, c_{t-1}, r_{t-1})$.

For every $c \in \mathbb{Z}_q$, there exists a polynomial f of degree at most $t - 1$ satisfies

$$\begin{aligned} &f(0) = c \\ &f(x_b) = c_b \quad for \quad b = \{1, \ldots, t-1\} \end{aligned}$$

It follows that players cannot recover the secret from $t-1$ shares. Furthermore, the subset of size $< t-1$ has fewer shares, so c also keeps private. □

In our protocol, $G_i^{c_i}$ and $H_i^{r_i}$ can be seen as the encryption of g^{c_i} and h^{r_i} respectively. As we pointed out early that it may be dangerous for our rational secret sharing protocol if players can break the encryption and get g^{c_i}. In fact, breaking the encryption is equivalent to the Diffie-Hellman problem.

Lemma 3. *Under the Diffie-Hellman assumption, it is infeasible to break the encryption of shares: $G_i^{c_i}$ and $H_i^{r_i}$.*

Proof. Given $G_i^{c_i} = g^{z_i \cdot c_i}$, $G_i = g^{z_i}$, breaking the encryption means computing g^{c_i} from $g^{z_i \cdot c_i}$ and g^{z_i}, which is equivalent to computing g^{β} from $g^{\alpha\beta}$ and g^{α}. If we can solve this problem, then given g^{α} and g^{β}, we can compute $g^{\beta^{-1}}$ from g^1 and g^{β}, so that we can compute $g^{\alpha\cdot\beta}$ from g^{α} and $g^{\beta^{-1}}$. This means that Diffie-Hellman problem can be settled. □

Our rational secret sharing protocol requires that g^c cannot be revealed before the reconstruction stage. The commitment $g^{c_i}h^{r_i}$ reveals no information of c_i and r_i, moreover, the above result shows that players except P_i cannot compute g^{c_i}, but it cannot guarantee that no information about g^{c_i} is revealed. We prove the following result which holds under the Decision Diffie-Hellman assumption.

Lemma 4. *Under the Decision Diffie-Hellman assumption, less than t players cannot get any information about g^c or h^r.*

Proof. Here, we prove that no information about g^c is revealed, the proof of h^r is similar. We start from subset of $t-1$ players, w.l.o.g. denoted $P_1, \ldots, P_{t-1}$, they can learn $g^{c_1}, \ldots, g^{c_{t-1}}$. If they can get some information about g^c then they can get partial information about g^{c_j} $(j > t-1)$ from its encryption $G_j^{c_j} = g^{z_j \cdot c_j}$. Writing $G_j^{c_j} = g^{z_j \cdot c_j} = g^{\alpha\cdot\beta}$, $G_j = g^{\alpha}$, we suppose that a player can determine whether the decrypted share g^{β} is equal to a given g^{δ} or not. In this situation, given g^{α}, g^{β}, g^{θ}, if the player can output whether $g^{\theta/\alpha}$ equals g^{β} with inputs g^{θ}, g^{α}, then we can determine whether g^{θ} equals $g^{\alpha\cdot\beta}$ or not. This is a contradiction with the Decision Diffie-Hellman assumption. □

Next we need to prove the consistency of shares in the PVSS scheme.

Lemma 5. *Under the Discrete Logarithm assumption, if all players accept their shares in the publicly verifiable secret sharing scheme, then their shares are consistent.*

Proof. According to [13], the dealer can succeed in distributing inconsistent shares unless he can settle Discrete Logarithm problem, so that the shares being committed are consistent. Moreover, following from Lemma 1, the discrete logarithm of $G_i^{c_i}$ equals the content c_i of $g^{c_i}h^{r_i}$. Thus, all players can make sure that the shares distributed are consistent. □

It follows from the Lemmas above that the commitments and encryptions do not reveal any information about the secret, and the shares held by players are consistent with the secret. We can get the result.