Constant-Round Rational Secret Sharing with Optimal Coalition Resilience

Akinori Kawachi[1] Yoshio Okamoto[2] Keisuke Tanaka[1]
Kenji Yasunaga[3]

[1] Tokyo Institute of Technology, Japan
{kawachi, keisuke}@is.titech.ac.jp
[2] The University of Electro-Communications, Japan
okamotoy@uec.ac.jp
[3] Kanazawa University, Japan
yasunaga@se.kanazawa-u.ac.jp

Abstract

We provide a general construction that converts any rational secret-sharing protocol to a protocol with a constant-round reconstruction. Our construction can be applied to protocols for synchronous channels, and preserves a strict Nash equilibrium of the original protocol. Combining with an existing protocol, we obtain the first rational secret-sharing protocol that achieves a strict Nash equilibrium with the optimal coalition resilience of $\lceil \frac{n}{2} \rceil - 1$, where n is the number of players.

Although the coalition resilience of $\lceil \frac{n}{2} \rceil - 1$ is known to be optimal for constant-round protocols, we show that the limitation can be circumvented by considering a natural assumption that players avoid reconstructing *fake* secrets. Under this assumption, we give a constant-round protocol that achieves a strict Nash equilibrium with the maximal coalition resilience of $n - 1$.

Our construction can be extended to a construction that preserves the *immunity* to unexpectedly behaving players. Then, we obtain a protocol that achieves a Nash equilibrium with the optimal coalition resilience of $\lceil \frac{n}{2} \rceil - t - 1$ in the presence of t unexpectedly behaving players for any constant $t \geq 1$.

1 Introduction

Recently, much attention has been paid to the interplay between game theory and cryptography. (For surveys, see [12, 19, 5, 10, 15].) One important research problem is to design cryptographic protocols for rational players in a game-theoretic sense. Traditionally, most cryptographic protocols have been designed for participants who are either honest or malicious. There is, however, no guarantee for non-malicious players to behave honestly in real-life situations. It is desirable that cryptographic protocols work for rational players.

Halpern and Teague [11] initiated the study of secret sharing for rational players, which is called *rational secret sharing*. The payoff function is characterized such that rational players prefer to learn the secret and prefer fewer players to learn the secret. Assuming this payoff function, it does not seem that conventional secret-sharing schemes work well. They proposed a rational secret-sharing protocol, in which the prescribed strategy is a Nash equilibrium and survives iterated elimination of weakly dominated strategies [11]. Since their work, rational secret-sharing protocols have been studied with various solution concepts, communication channels, and characteristics of players [9, 1, 16, 13, 14, 20, 18, 2, 7, 21] together with several impossibility results [11, 14, 2].

Some previous protocols on rational secret sharing [11, 9, 1, 16, 13, 14, 7] are essentially based on the same underlying idea: The protocol consists of real rounds and fake rounds. Players can learn the secret only in real rounds and cannot learn any information in fake rounds. Since the probability of being a real round is sufficiently small and the protocol halts if some player was silent in fake rounds, the best action players can take is to reveal their shares in every round. As long as relying on this general idea, it seems difficult to construct constant-round protocols since the players must perform in fake rounds to learn the secret. Indeed, the round complexity of the protocols in [11, 9, 13, 14, 7] is $O(1/\beta)$, where β is a sufficiently small value depending on the payoff values. Since Shamir's original secret-sharing scheme [24] takes only one round to learn the secret, it is desirable that the round complexity be a small constant independent of the payoff values. Among the previous work, only the protocols in [1, 18, 20, 2] achieve constant-round reconstruction.

In reconstructing the secret, several rational secret-sharing protocols [11, 9, 1, 16, 13, 2] require the involvement of the dealer or, to eliminate the on-line dealer, general multi-party computation. Also, many rational secret-sharing protocols [11, 9, 1, 16, 13, 14, 2] require a simultaneous broadcast channel, which is a relatively strong assumption of the communication channel. Thus, it is desirable to design constant-round rational secret-sharing protocols with low computational cost without the on-line dealer or simultaneous channels.

1.1 Our Results

We present a general construction of a rational secret-sharing protocol. Our construction employs an existing rational secret-sharing protocol as a sub-protocol in a black-box manner. In the protocol obtained by our construction, players can reconstruct the secret in expected three rounds. Moreover, the players can learn the secret *without using the shares of the sub-protocol with high probability*. Thus, even if the round/computational complexity of the reconstruction of the sub-protocol is large, our construction converts such a protocol to a protocol with efficient reconstruction. The communication channel we assume is a synchronous broadcast channel, which is used in [13, 14, 2, 7] and strictly weaker than a simultaneous broadcast channel, which is broadly used in previous work [11, 9, 1, 16, 13, 14, 2]. Our construction works well in both of information-theoretic and computational settings.

Strong solution concepts. For any $3 \leq t \leq n$, our construction yields a t-out-of-n secret-sharing protocol that achieves a *strict* Nash equilibrium if so is the sub-protocol. A strict Nash equilibrium, of which an information-theoretic version was introduced in [14] and a computational version was in [7], is preferable to a plain Nash equilibrium. While a Nash equilibrium only guarantees that deviations do not increase the payoff, a strict Nash equilibrium guarantees that any deviation strictly decreases the payoff. We also prove that a strict Nash equilibrium implies another preferable equilibrium called a *Nash equilibrium that is stable with respect to trembles*, which was introduced in [7]. Intuitively, the stability with respect to trembles guarantees that even if a player believes that the other players might follow any strategy other than the prescribed one with small probability, there is no better strategy for the player than the prescribed one. Our implication shows that a strict Nash equilibrium is a relatively strong solution concept that captures the stability against any small deviations of players.

Optimal coalition resilience. Our construction preserves the coalition resilience of the underlying sub-protocol. Protocols resilient to deviations by coalitions have been studied in several previous work [1, 13, 2, 7]. In our construction, if the sub-protocol achieves a strict Nash equilibrium with a coalition resilience of $\lceil \frac{t}{2} \rceil - 1$, then the resulting protocol achieves a strict Nash equilibrium with the same coalition resilience. The resilience of $\lceil \frac{t}{2} \rceil - 1$ is optimal for constant-round t-out-of-n protocols. The optimality can be shown by almost the same argument as Asharov and Lindell [2] who proved that no constant-round n-out-of-n protocol can achieve a Nash equilibrium with a coalition resilience of $\lceil \frac{n}{2} \rceil$.

By plugging the protocol of Fuchsbauer, Katz, and Naccache [7] into our construction, we obtain an expected constant-round t-out-of-n secret-sharing protocol that achieves a strict Nash equilibrium with the optimal coalition resilience of $\lceil \frac{t}{2} \rceil - 1$. Note that the protocol of [7] achieves a coalition resilience of $t-1$, and the expected round complexity is $O(1/\beta)$ for some small β depending on the payoff values. As far as we know, there was no constant-round protocol that achieves a strict Nash equilibrium in non-simultaneous channels.

Circumventing the impossibility result of Asharov and Lindell [2]. As described above, Asharov and Lindell [2] showed the limitation of coalition resilience of constant-round protocols. We show that the limitation can be circumvented by considering a natural assumption on the payoff functions. Specifically, we assume that players avoid learning *fake* secrets. As far as we know, such an assumption has not been considered in the previous studies. Then, a slightly modified version of our construction provides a protocol that achieves a strict Nash equilibrium with the maximal coalition resilience of $n - 1$ for any $n \geq 2$.

Immunity to unexpectedly behaving players. Furthermore, we also provide a general construction that preserves the *immunity* to "unexpectedly behaving" (or malicious) players. The immunity is also considered in [1, 16], and guarantees certain robustness of protocols. In particular, the immunity is a desirable property for protocols conducted by rational players since such protocols rely on the rationality of players, but it seems difficult to understand the rationality of every player precisely. If a protocol does not have the immunity, the protocol would not work well even if only a single player behaves unexpectedly. Indeed, several existing protocols including our protocol described above do not work in the presence of such players. Thus, we construct a *non-constant* round protocol that achieves a Nash equilibrium with a coalition resilience of $\lceil \frac{n}{2} \rceil - t$ in the presence of t malicious players for a constant t. The protocol is a variant of the protocol of [7]. By applying our general construction to this protocol, we obtain a constant-round protocol that achieves a Nash equilibrium with the optimal coalition resilience of $\lceil \frac{n}{2} \rceil - t$ in the presence of t malicious players for a constant t. Our general construction with immunity requires symmetric-key encryption and pseudorandom functions, and thus fits the computational setting. We also discuss the optimality of our protocols with immunity regarding the coalition resilience.

1.2 Our Approach

We describe the idea of our general construction. The dealer chooses conventional secret-sharing schemes S_1 and S_2, and a rational secret-sharing protocol S_3. The secret is shared by S_1 and S_3, but with small probability, the secret for S_1 is fake. The information on whether the secret of S_1 is real or fake is shared by S_2. In the reconstruction, first, players are requested to reveal the shares of S_1, and then proceed to the next round only if *all* the players have honestly revealed the shares. In the first round, all the players have an incentive to reveal their shares. Let t_1 be the threshold of the secret sharing scheme S_1. Then, the t_1-th sender in this round can reconstruct the secret by using her own share. Nevertheless, she will reveal her share since the secret may be fake, and if she did not reveal, the protocol halts along with the possibility that the secret is fake. The small probability that the secret is fake poses a "threat" to players so as to reveal their shares in the first round. In the second round, all the players are requested to reveal the shares of S_2, and then proceed to the next round only if *all* the players have honestly revealed the shares *and* the reconstructed secret is *fake*. Let t_2 be the threshold of the secret sharing scheme S_2. Then, the t_2-th sender in this round will learn whether the secret is real or fake by using her own share. If the secret is fake, she reveals her share since if not, the protocol does not proceed to the next round and she cannot learn the secret. If the secret is real, then she has no incentive to participate in the protocol, and thus she may not reveal her share, but this action *signals* to the other players that the secret is real. Therefore, every player can recognize that the reconstructed secret in the previous round is real. If no player deviates in the first two rounds and the reconstructed secret is fake, all the players are guaranteed to learn the secret eventually by S_3. The idea of using the secret sharing S_2 as the signal indicating whether a reconstructed secret of S_1 is real or fake is similar to the idea used in [7] for achieving a strict Nash equilibrium for non-simultaneous channels.

1.3 Related Work

Constant-round reconstruction for rational secret sharing was achieved by Abraham, Dolev, Gonen, and Halpern [1], Micali and shelat [18], Ong, Parkes, Rosen, and Vadhan [20], and Asharov and Lindell [2]. The protocols in [1, 2] achieve a Nash equilibrium with a coalition resilience of $\lceil \frac{n}{2} \rceil - 1$. Their protocols employ as a sub-protocol another rational secret-sharing protocol such as [9, 1, 13, 14, 7]. Their protocols assume a simultaneous broadcast channel and need to perform the sub-protocol to reconstruct the secret several times. Our construction also requires another protocol as a sub-protocol, but the sub-protocol does not actually perform with high probability. Thus our reconstruction has low computational cost even if the sub-protocol needs high computational cost.

Additionally, our construction can employ any protocol achieving a strict Nash equilibrium in a *black-box* manner, while the protocols in [1, 2] need a certain type of protocols as a sub-protocol. A black-box construction in the literature of rational cryptography has appeared in the study on a novel framework called *rational protocol design* [8], where designing protocols itself is modeled as a game between a protocol designer and an attacker. While Garay et al. [8] show general composition theorems in their framework, as far as we know, black-box constructions have not appeared in the literature of rational secret sharing.

The protocol of [18] achieves a stronger solution concept than other work including ours. However, their protocol requires a stronger communication channel than the others. The protocol of [20] is quite efficient with respect to both round complexity and computational cost with a weaker communication channel. However, a sufficient number of honest players are assumed to exist. Namely, while these protocols satisfy desirable properties, they require a stronger assumption than ours in some sense.

2 Models and Settings

2.1 Secret-Sharing Scheme

A t-out-of-n secret-sharing scheme consists of two phases: the sharing phase and the reconstruction phase. In the sharing phase, the dealer holds the secret and distributes shares of the secret to n parties called *players*. In the reconstruction phase, the players reconstruct the secret from their shares. We consider the two requirements, correctness and secrecy. The correctness guarantees that any subset of $t^* \geq t$ players can reconstruct the secret if they perform the reconstruction phase honestly. The secrecy guarantees that any subset of $t^* < t$ players can learn nothing about the secret.

2.2 Secret-Sharing Reconstruction Game

We assume that the players are rational. In secret-sharing schemes, the reconstruction phase can be considered as a game for the players. Therefore, we see the reconstruction protocol as a pair of a game and a prescribed strategy for the game. The goal is to design a protocol that will result in the desired outcome: all the participants reconstruct the secret. We say a (rational) secret-sharing protocol Π is *t-out-of-n* if, in addition to the t-out-of-n property of secret sharing, the shares are distributed to n players, and the reconstruction protocol can be performed in the presence of $t^* \geq t$ players. We say that Π is a t-out-of-n secret-sharing "protocol" when we see it as rational secret-sharing, and that Π is a t-out-of-n secret-sharing "scheme" when we see it as a conventional secret-sharing scheme.

Following [7], we model a reconstruction game in a way such that a secret s is chosen uniformly at random from the domain of secrets, and every player finally outputs some value, which the player wants to be the same as the secret s. The advantages of modeling a game in this way are discussed in [7]. Let $N = \{1, \ldots, n\}$ be the set of players in the protocol. The outcome of the game is denoted by $o = (o_1, o_2, ..., o_n)$, where o_i is a random variable that equals 1 if the output of player i is s, and 0 otherwise. The payoff of each player is determined by the outcome of the game.

The tuple of strategies $\sigma = (\sigma_1, \ldots, \sigma_n)$ is called a *strategy profile* for the game, where σ_i corresponds to the strategy for player i. Let C be a subset of N. We define σ_C to be the tuple of strategies σ_i for $i \in C$. Following the game-theoretic notation, we define $\sigma_{-i} = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$, and $\sigma_{-C} = \sigma_{N \setminus C}$. For two strategy profiles σ and $\sigma' = (\sigma'_1, \ldots, \sigma'_n)$, we write $(\sigma'_i, \sigma_{-i}) = (\sigma_1, \ldots, \sigma_{i-1}, \sigma'_i, \sigma_{i+1}, \ldots, \sigma_n)$ for $i \in N$, and write (σ'_C, σ_{-C}) as the strategy profile in which the strategy of player i is σ'_i if $i \in C$ and σ_i otherwise. If all actions specified in each σ_i can be computed by a probabilistic polynomial-time Turing machine, the strategy profile σ is called PPT. Let $u_i(\sigma)$ denote the expected payoff of player i when the players follow a strategy profile σ.

2.3 Game-Theoretic Notions

For a secret-sharing protocol Π, if the prescribed strategy σ of the reconstruction protocol Π achieves a game-theoretic concept, say A, in the reconstruction game, we say that Π *induces* A.

The definitions of computational Nash equilibrium, computational strict Nash equilibrium, stability with respect to trembles (which appears in Appendix A), and their extension to coalition resilience follow the definitions of [7]. Regarding the immunity to malicious players, we provide a computational definition which is an extension of the information-theoretic definition of [1]. Hereafter, we use k as the security parameter of cryptographic primitives. We say a function $\epsilon : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every sufficiently large k, $\epsilon(k) < k^{-c}$ for any constant c. A function $\delta : \mathbb{N} \to \mathbb{R}$ is called *noticeable* if $\delta(\cdot)$ is not negligible.

Definition 1 (Computational Nash equilibrium)**.** *A PPT strategy profile σ is a* computational Nash equilibrium *if for any $i \in N$ and any PPT strategy σ'_i for player i, $u_i(\sigma'_i, \sigma_{-i}) \leq u_i(\sigma) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function.*

To define a strict Nash equilibrium, we use the notion of equivalent play, which was introduced in [7]. Let σ be a prescribed strategy profile and σ'_i a strategy for player i. Define the view of player i in the game to include the information given by the dealer to player i, the random coin of player i, and all the messages received from player $j \neq i$, but not including any messages from player j after player i writes to her (write-once) output tape. We say the strategy σ'_i *yields equivalent play with respect to* σ, denoted by $\sigma'_i \in_{\mathsf{eq}} \sigma$, if, given the views of all the players $j \neq i$ in which they follow σ_{-i} and player i follows either σ_i or σ'_i, no PPT machine can distinguish whether player i follows σ_i or σ'_i. (See [7] for the detailed definition.) We write $\sigma'_i \notin_{\mathsf{eq}} \sigma$ if σ'_i does not yield equivalent play with respect to σ.

A strict Nash equilibrium guarantees that if some player deviates from the prescribed protocol, then the payoff of the player decreases by some noticeable amount.

Definition 2 (Computational strict Nash equilibrium)**.** *A PPT strategy profile σ is a* computational strict Nash equilibrium *if (1) σ is a computational Nash equilibrium; and (2) for any $i \in N$ and any PPT strategy σ'_i for which $\sigma'_i \notin_{\mathsf{eq}} \sigma$, there is a constant $c > 0$ such that $u_i(\sigma) \geq u_i(\sigma'_i, \sigma_{-i}) + k^{-c}$ for infinitely many $k > 0$.*

The above definitions consider deviations by a single player. We also consider deviations of a coalition of players. A *coalition* is a subset of the players N. We consider a coalition is a set of players who may be coordinated by some single party. Therefore, we assume that each coalition has one payoff function. Let $C \subset N$ be a coalition and $u_C(\cdot)$ the payoff function of C.

Definition 3 (Coalition resilience)**.** *A PPT strategy profile σ is an* r-resilient computational Nash equilibrium *if for any $C \subset N$ with $|C| \leq r$ and any PPT strategy σ'_C, $u_C(\sigma'_C, \sigma_{-C}) \leq u_C(\sigma) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function.*

We also define the coalition-resilient variant of strict Nash equilibrium. For $C \subset N$, we write $\sigma'_C \subset_{\mathsf{eq}} \sigma$ if $\sigma'_i \in_{\mathsf{eq}} \sigma$ for all $i \in C$, and $\sigma'_C \not\subset_{\mathsf{eq}} \sigma$ if $\sigma'_C \notin_{\mathsf{eq}} \sigma$ for some $i \in C$.

Definition 4. *A PPT strategy profile σ is an* r-resilient computational strict Nash equilibrium *if (1) σ is an r-resilient computational Nash equilibrium; and (2) for any $C \subset N$ with $|C| \leq r$ and any PPT strategy σ'_C for which $\sigma'_C \not\subset_{\mathsf{eq}} \sigma$, there is a constant $c > 0$ such that $u_C(\sigma) \geq u_C(\sigma'_C, \sigma_{-C}) + k^{-c}$ for infinitely many $k > 0$.*

We introduce the notion of immunity to unexpectedly behaving players. The immunity guarantees that even if some players behave unexpectedly (or maliciously) in the game, the behavior does not affect the payoffs of the other players.

Definition 5 (Immunity)**.** *A PPT strategy profile σ is* computationally t-immune *if for any $T \subset N$ with $|T| \leq t$, any PPT strategy σ'_T, and any player $i \notin T$, $u_i(\sigma) \leq u_i(\sigma_{-T}, \sigma'_T) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function.*

Let $\{A, B, C\}$ be a partition of N. For strategy profiles σ, ρ, ϕ, we write $(\sigma_A, \rho_B, \phi_C)$ as the strategy profile in which the strategy of player i is σ_i if $i \in A$, ρ_i if $i \in B$, and ϕ_i if $i \in C$. We define a combination of coalition resilience and immunity. The following is a computational version of the definition of robustness defined in [1].

Definition 6. *A PPT strategy profile σ is an* (r, t)-robust computational Nash equilibrium *if (1) for any $C, T \subset N$ such that $C \cap T = \emptyset$, $1 \leq |C| \leq r$, and $0 \leq |T| \leq t$, any PPT strategy ρ_T, and any PPT strategy σ'_C, we have $u_C(\sigma_{N \setminus (C \cup T)}, \sigma'_C, \rho_T) \leq u_C(\sigma_{-T}, \rho_T) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function, and (2) σ is computationally t-immune.*

The (r, t)-robustness guarantees that even if at most t players behave maliciously, the strategy is still an r-resilient Nash equilibrium, and that the malicious behavior does not affect the payoffs of the other players. Note that an (r, t)-robustness implies both an r-resilient Nash equilibrium and t-immunity. This is because an r-resilient Nash equilibrium is a special case of the first condition that $t = 0$, and t-immunity appears as the second condition. We can also define an (r, t)-robust computational *strict* Nash equilibrium analogously.

2.4 Payoff Functions of Players

The payoff function of players in reconstruction games follows the previous studies. First, players prefer to learn the secret. Second, players prefer fewer players to learn the secret. The payoffs for a secret-sharing reconstruction game depend only on the outcome of the game. We write $u_i(o)$ as the payoff of player i for the outcome o. For two outcomes o and o', we assume that (1) if $o_i > o'_i$, then $u_i(o) > u_i(o')$; (2) if $o_i = o'_i$ and $\sum_{j \in N} o_j < \sum_{j \in N} o'_j$, then $u_i(o) > u_i(o')$. In our analysis, we need the following values of the payoff.

1. $u_i(o) = U^+$ if player i learns the secret and no other player does.
2. $u_i(o) = U$ if all the players in the reconstruction game learn the secret.
3. $u_i(o) = U^-$ if player i does not learn the secret.

It should hold that $U^+ \geq U > U^-$. We assume that there is a noticeable function $\delta_1(\cdot)$ such that $U \geq U^- + \delta_1(k)$, where k is the security parameter. We define $U_{\text{random}} = \frac{1}{|\mathcal{S}|} \cdot U^+ + \left(1 - \frac{1}{|\mathcal{S}|}\right) \cdot U^-$, where $\mathcal{S}$ is the domain of secrets in the secret-sharing protocol. The value U_{random} is the payoff of a player who outputs a random guess for the secret assuming that the other parties halt without any output or with the wrong outputs. We also assume that there is a noticeable function $\delta_2(\cdot)$ such that $U \geq U_{\text{random}} + \delta_2(k)$. This inequality means that players have an incentive to perform the reconstruction protocol.

Regarding the payoff of coalitions, we also follow the formalization of [7]. We assume the coalition C outputs a single value in a game. The outcome of a game with the coalition C consists of the outcome o_i of player $i \in N \setminus C$ and the outcome o_C of the coalition C. The outcome o_C takes 1 if C outputs the secret, and 0 otherwise. Let $u_C(o)$ denote the payoff of the coalition C for the outcome o of the game with the coalition C. Then, we define

1. $u_C(o) = U^+$ if C learns the secret and no player outside C does.
2. $u_C(o) = U$ if all the players in the reconstruction game learn the secret.
3. $u_C(o) = U^-$ if C does not learn the secret.

It should hold that $U^+ \geq U > U^-$. We also define U_{random} as in the case of a single deviation. Then, we assume that there are noticeable functions $\delta_3(\cdot)$ and $\delta_4(\cdot)$ such that $U \geq U^- + \delta_3(k)$ and $U \geq U_{\text{random}} + \delta_4(k)$.

2.5 Communication Channels

We assume that players can use only a synchronous but non-simultaneous broadcast channel [13, 14]. With this channel, the protocol proceeds in rounds, and each round consists of n sub-rounds for each player, where n is the number of players in the protocol. In each sub-round, only a single player can send a message. We assume that if a player does not send any message (within some predetermined time), the other players will receive the special symbol $\perp$ from her.

3 Our Protocols

In this section, we present our general constructions for rational secret-sharing protocols. For the simplicity of the explanation, we first present the n-out-of-n protocol in Section 3.1. A general t-out-of-n protocol is presented in Appendix B.1 as a generalization of the n-out-of-n protocol. In Section 3.2, we show that the limitation of coalition resilience proved in [2] can be circumvented by considering players who do not prefer to reconstruct fake secrets. In Section 3.3, we provide a construction of protocols with immunity to malicious players, and discuss the optimality of the protocols regarding the coalition resilience.

3.1 The n-out-of-n Protocol

Our protocol proceeds as follows. The dealer chooses conventional $(\lfloor \frac{n}{2} \rfloor + 1)$-out-of-$n$ secret sharing scheme S_1 and $\lceil \frac{n}{2} \rceil$-out-of-$n$ secret sharing scheme S_2, and a rational secret sharing protocol S_3. For a (random) secret s, the dealer generates shares of S_1 and S_3, but with small probability α, the secret of S_1 is a fake secret, which is chosen uniformly at random and thus indistinguishable from the real secret s.[1] The information whether the secret of S_1 is real or fake is secret-shared by S_2. Then the shares of S_1, S_2, and S_3 are distributed to the players. In the reconstruction phase, first the players are requested to reveal their shares of S_1, and then proceed to the next round only if *all* the players revealed their valid shares in this round. In the second round, the players are requested to reveal their shares of S_2 to learn whether the reconstructed secret of S_1 is real or fake. Then, the players proceed to the reconstruction protocol of S_3 only if the secret is *fake* and *all* the players revealed their valid shares in the second round.

We describe an intuition that our protocol achieves a strict Nash equilibrium. In our protocol, any deviation from the protocol decreases the payoff of players as long as coalitions of size at most $\lceil \frac{n}{2} \rceil - 1$ are considered. Note that, in the analysis of the strict Nash equilibrium, we can assume that the players outside a coalition follow the prescribed strategy. In the first round, since the

[1] Although the existence of a fake secret that is indistinguishable from the real secret may be a strong requirement, this requirement is inevitable to construct a fair protocol for non-simultaneous channels [2].

number of players in the coalition is at most $\lceil \frac{n}{2} \rceil - 1$, at least $n - (\lceil \frac{n}{2} \rceil - 1) = \lfloor \frac{n}{2} \rfloor + 1$ valid shares will be revealed. Thus, all the players can learn the secret of S_1 regardless of the actions of the coalition. If some player in the coalition revealed an invalid share in the first round, the players will not proceed to the next round (or equivalently, the protocol will halt) along with the possibility that the reconstructed secret is fake, which decreases the payoff. In the second round, the first $\lceil \frac{n}{2} \rceil - 1$ players (outside the coalition), who do not know whether the reconstructed secret is real or fake, will reveal a valid share. This is because if any invalid share is revealed, the players will not proceed to the third round and thus there remains the possibility that the real secret cannot be reconstructed. After receiving the $\lceil \frac{n}{2} \rceil - 1$ shares, the rest of players can verify that the secret is real or fake by using her own share. If the secret is fake, the rest of players also reveal their valid shares in order to go to the next round for reconstructing the real secret using the shares of S_3. If the secret is real, the rest of players may not reveal their valid shares, but this signals to the other players that the secret is real, and thus all the players can learn the secret. However, to achieve a strict Nash equilibrium, in which any deviation implies the decrease of the payoff, we cannot allow any deviation when all the players can learn the secret. Thus, in the protocol, we allow players to be silent (or reveal invalid shares) in this round if the reconstructed secret is real. This implies that in this round deviations occur only when the reconstructed secret is fake.

To achieve an n-out-of-n property, we employ the masking technique used in [2] at the cost of one additional round in the reconstruction phase. In the sharing phase, the dealer chooses μ uniformly at random and masks the secret s by taking $\mu \oplus s$. Then the above protocol is performed in which $\mu \oplus s$ is considered as the secret. The mask μ is shared by a conventional n-out-of-n secret-sharing scheme S_0. In the reconstruction phase, players first are requested to reveal their shares of S_0 to reconstruct μ. If some player deviates, the protocol halts and no player can learn the secret.

In order to check the validity of the received shares, we use *authenticated* secret-sharing to share the secret. Thus, players can generate invalid shares only with a negligible probability. The definition of an authenticated secret-sharing scheme is provided in Appendix C.1.

The protocol employs an n-out-of-n authenticated secret-sharing scheme S_0, an $(\lfloor \frac{n}{2} \rfloor + 1)$-out-of-$n$ authenticated secret-sharing scheme S_1, an $\lceil \frac{n}{2} \rceil$-out-of-$n$ authenticated secret-sharing scheme S_2, and an n-out-of-n rational secret-sharing protocol S_3. Note that, in the protocol below, we can choose the probability α to be k^{-c} for *any* constant c, where k is the security parameter.

Sharing phase. To share a secret $s \in \{0,1\}^\ell$, the dealer performs the following:

- Choose $\mu \in \{0,1\}^\ell$ uniformly at random, and generate shares $(w_1, \ldots, w_n)$ of S_0 with the secret μ.
- Set $s' = \begin{cases} \mu \oplus s & \text{with probability } 1-\alpha, \\ \mathsf{fake} & \text{with probability } \alpha, \end{cases}$, where $\mathsf{fake} \in \{0,1\}^\ell$ is chosen uniformly at random, and generate shares $(x_1, \ldots, x_n)$ of S_1 with the secret s'.
- Set $s'' = 1$ if $s' = \mu \oplus s$ in the previous step, and $s'' = 0$ otherwise, and generate shares $(y_1, \ldots, y_n)$ of S_2 with the secret s''.
- Generate shares $(z_1, \ldots, z_n)$ of S_3 with the secret s.
- Send (w_i, x_i, y_i, z_i) to player $i \in N$.

Reconstruction phase. After all the players received the shares, the players perform the following:

- For all $i \in N$ (in any order), send w_i.

  After all the players broadcasted their messages, if all the shares are valid, reconstruct μ from $(w_1, \ldots, w_n)$ and go to the next round. Otherwise, halt and output a random string in $\{0,1\}^\ell$.

- For all $i \in N$ (in any order), send x_i.
  - After all the players broadcasted their messages, set N^* to be the set of players $j \in N$ who sent the valid share. If $|N^*| \geq \lfloor \frac{n}{2} \rfloor + 1$, reconstruct s' from $(x_1, \ldots, x_n)$. Otherwise, set s' to a random string in $\{0,1\}^\ell$.
  - If $|N^*| = n$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- For all $i \in N$ (in any order), send y_i. (Each player is allowed to take any action when $s'' = 1$.)
  - After all the players broadcasted their messages, update N^* to be the set of players $j \in N^*$ who sent the valid share. If $|N^*| \geq \lceil \frac{n}{2} \rceil$, reconstruct s'' from $(y_1, \ldots, y_n)$.
  - If $|N^*| = n$ and $s'' = 0$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- Perform the reconstruction protocol of S_3 by using z_i to reconstruct s.

  Then, halt and output s.

Theorem 1. *For any $n \geq 3$, the above is an n-out-of-n secret-sharing protocol that induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium if S_3 induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium. The secret is reconstructed in three rounds with probability at least $1 - k^{-c}$, and the expected number of rounds for reconstruction is at most $3 + \tau \cdot k^{-c}$ for any constant c, where k is the security parameter and τ is the expected number of rounds for reconstruction in S_3.*

Proof: Note that α can be chosen to be k^{-c} for any constant c. The condition $n \geq 3$ comes from the fact that we need a non-trivial coalition-resilient Nash equilibrium for S_3, namely, $\lceil \frac{n}{2} \rceil - 1 \geq 1$.

First, we claim that our protocol has the n-out-of-n property. Since the secret s is masked by μ, which is shared by the n-out-of-n secret-sharing S_0, at most $n-1$ shares of $\{w_i\}$ reveal no information on s. Also, at most $n-1$ shares of $\{z_i\}$ reveal no information on s since each z_i is a share of the n-out-of-n secret sharing S_3. The correctness of the protocol follows from the correctness of the underlying schemes S_0, S_1, S_2, and S_3. If all the n players follow the protocol, they can learn the secret in the third round with probability $1 - \alpha$, and in the later rounds with probability α. Note that, although we allow players to take any action when the reconstructed secret is real in the third round, this does not affect the fact that the players can learn the secret in the third round if they follow the protocol. Therefore, the secret is reconstructed in three rounds with probability $1 - \alpha = 1 - k^{-c}$, and the expected number of rounds for reconstruction is $3(1-\alpha) + \tau \cdot \alpha \leq 3 + \tau \cdot k^{-c}$.

Next, we prove that the protocol induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium. In the analysis, we assume that, when a player is requested to send a share of authenticated secret-sharing schemes, all the actions that the player can take are sending the valid share and being silent. This is because sending an invalid share is regarded as being silent and the probability of successfully generating another valid share is negligible, which follows from the authenticity of authenticated secret-sharing schemes.

Let $C \subset N$ be any coalition with $|C| \leq \lceil \frac{n}{2} \rceil - 1$. Let σ be the prescribed strategy of the protocol. Then it follows from the correctness of the protocol that $u_C(\sigma) = U$. First, we show that σ is a computational Nash equilibrium. Namely, for any strategy σ'_C of C, we show that $u_C(\sigma'_C, \sigma_{-C}) \leq U + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$. Note that, in evaluating the value of $u_C(\sigma'_C, \sigma_{-C})$, we can assume that the players in $N \setminus C$ follow the prescribed strategy σ.

In the first round, if some player in C is silent, then the players in $N \setminus C$ do not proceed to the later rounds. The shares $\{w_i\}_{i \in N}$ only contain the information on μ. Also, the shares $\{(x_i, y_i, z_i)\}_{i \in C}$ reveal no information on s' or s since the thresholds of S_1, S_2, and S_3 are strictly greater than $\lceil \frac{n}{2} \rceil - 1 \geq |C|$. Thus, the coalition C cannot learn the secret s. Therefore, the payoff of C is at most $\max\{U^-, U_{\text{random}}\}$, which is noticeably less than U.

In the second round, every player in N can reconstruct s' regardless of the strategy of C because the players in $N \setminus C$ reveal their valid shares, and thus the number of valid shares revealed is at least $|N \setminus C| \geq n - (\lceil \frac{n}{2} \rceil - 1) = \lfloor \frac{n}{2} \rfloor + 1$, which is at least the threshold $\lfloor \frac{n}{2} \rfloor + 1$ of S_1. Note that, at this point, the coalition C does not learn whether s' is real or fake. This is because s and s' are indistinguishable since both s and s' are distributed uniformly at random, and s'' is shared by $\lceil \frac{n}{2} \rceil$-out-of-$n$ secret sharing. If some player in C is silent in the second round, then the players in $N \setminus C$ do not proceed to the later rounds, and thus the coalition C cannot learn whether s' is real or fake. Still, the coalition C cannot learn the secret s from $\{z_i\}_{i \in C}$. Therefore, since $s' = \mathsf{fake}$ with probability α, the expected payoff of C is $u_C(\sigma'_C, \sigma_{-C}) \leq (1 - \alpha) \cdot U + \alpha \cdot \max\{U^-, U_{\text{random}}\}$, which is noticeably less than U since $\alpha = k^{-c}$ for a constant c.

In the third round, if some player in C is deviated from the protocol, which means that $s'' = 0$, namely, $s' = \mathsf{fake}$, the players in $N \setminus C$ do not proceed to the later rounds. Then, the coalition C cannot learn the secret s since the only way to learn s is to use $\{z_i\}_{i \in C}$, but they are the shares of the n-out-of-n secret sharing S_3. Therefore, the payoff of C is $u_C(\sigma'_C, \sigma_{-C}) \leq \max\{U^-, U_{\text{random}}\}$, which is noticeably less than U.

If no player in C has deviated in the first three rounds and $s' = \mathsf{fake}$, the players will go to the reconstruction protocol of S_3. Since S_3 induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium, the expected payoff of C is at most $U + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$ if players in C deviated from the protocol of S_3.

Therefore, in any case, the expected payoff is $u_C(\sigma'_C, \sigma_{-C}) \leq U + \epsilon(k)$ for any strategy σ'_C of C, and thus the protocol induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational Nash equilibrium.

To complete the proof of the computational strict Nash equilibrium, we need to show that for any strategy σ'_C of C such that $\sigma'_C \notin_{\mathsf{eq}} \sigma$, $u_C(\sigma'_C, \sigma_{-C}) \leq u_C(\sigma) - k^{-c'} = U - k^{-c'}$ for a constant c'. The proof follows from the above analysis along with the fact that in the first three rounds, each player has a unique valid share she can send. If the strategy $\sigma'_C \notin_{\mathsf{eq}} \sigma$ is such that players in C deviate from the protocol in the first three rounds, then it follows from the above analysis that $u_C(\sigma'_C, \sigma_{-C}) \leq U - k^{-c'}$ for some constant c'. If σ'_C is such that players in C follow the protocol in the first three rounds, but deviate in the fourth or later rounds, since S_3 induces a strict Nash equilibrium, we have that $u_C(\sigma'_C, \sigma_{-C}) \leq U - k^{-c'}$ for some constant c'. Therefore, the protocol induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium. □

Note that our protocol can use an information-theoretic rational secret-sharing as a sub-protocol. Then, the resulting protocol induces an information-theoretic strict Nash equilibrium in a sense of [14] if the sub-protocol induces a strict Nash equilibrium.

Our protocol achieves another strong solution concept, a Nash equilibrium that is stable with respect to trembles [7], since we can prove that an r-resilient strict Nash equilibrium implies an r-resilient Nash equilibrium that is stable with respect to trembles. We provide the formal definition and theorem in Appendix A.

Although the above protocol also works in simultaneous broadcast channels as well, we can construct a simpler protocol for simultaneous channels. In particular, the expected round complexity is two in the resulting protocol. The description of the protocol is presented in Appendix B.2.

3.2 Circumventing the Impossibility Result of Asharov and Lindell [2]

As described in Section 1.1, Asharov and Lindell [2] show the limitation of coalition resilience of constant-round protocols. Namely, they show that no n-out-of-n secret-sharing protocol can achieve a Nash equilibrium with a coalition of $\lceil \frac{n}{2} \rceil$. In this section, we show that we can circumvent their limitation by assuming that players avoid learning *fake* secrets. Although this assumption seems reasonable, such an assumption on the payoff function has not been considered in the previous studies. A similar payoff function for which players obtain a high payoff when another player learn fake secrets was considered in [2].

We describe an intuition that we can achieve a coalition resilience of $n-1$ under this assumption. Let t be the threshold of the secret sharing scheme S_1 in the protocol described in Section 3.1. In the round for reconstructing the secret of S_1, after the first $t-1$ players broadcasted their valid shares, the rest of the players can reconstruct the secret s by themselves. However, the secret of S_1 is fake with some probability α. Thus, the players who do not want to obtain fake secrets will not output the reconstructed secret. Therefore, the rest of the players also reveal their valid shares and proceed to the next round to eliminate the possibility of being a fake secret.

We can construct an n-out-of-n rational secret sharing protocol with coalition resilience of $n-1$ by modifying the protocol described in Theorem 1 such that both secret-sharing schemes S_1 and S_2 are replaced with n-out-of-n secret-sharing schemes.

Theorem 2. *For any $n \geq 2$, the modified protocol is an n-out-of-n secret-sharing protocol that induces an $(n-1)$-resilient computational strict Nash equilibrium if S_3 induces an $(n-1)$-resilient computational strict Nash equilibrium and it holds that*

$$\frac{U - \max\{U^-, U_{\mathrm{random}}\}}{U^+ - U} \geq k^d \tag{1}$$

for some constant $d > 1$, where k is the security parameter. The secret is reconstructed in three rounds with probability at least $1 - k^{-c}$, and the expected number of rounds for reconstruction is at most $3 + \tau \cdot k^{-c}$ for any constant c such that $c \leq d$, where τ is the expected number of rounds for reconstruction in S_3.

Proof: We choose α to be k^{-c} for any constant c such that $c \leq d$. The n-out-of-n property follows from the same argument as the proof of Theorem 1.

We prove that the protocol induces an $(n-1)$-resilient computational strict Nash equilibrium if equation (1) holds and S_3 induces an $(n-1)$-resilient computational strict Nash equilibrium.

Let $C \subset N$ be any coalition with $|C| \leq n-1$. Let σ be the prescribed strategy of the protocol. We show that σ is a computational Nash equilibrium. Namely, for any strategy σ'_C of C, we show that $u_C(\sigma'_C, \sigma_{-C}) \leq U + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$. Since the coalition C is of size at most $n-1$, we can assume that at least one player in N honestly follows the protocol. Let i^* be such a player. Namely, $i^* \in N \setminus C$.

If some player in C is silent in the first round, by the same argument as in the proof of Theorem 1, we can show that the payoff of C is at most $\max\{U^-, U_{\text{random}}\}$, which is noticeably less than U.

Next, we consider the maximum expected payoff of C in the case that some player in C is silent in the second round. The maximum expected payoff is realized if player i^* is not the last sender in the second round. In that case, if all but the last sender in C send the valid shares, the coalition C can learn the secret s' but player i^* does not. However, since player i^* does not proceed to the later rounds, the coalition C cannot learn whether s' is real or fake, and indeed s' is fake with probability α. Then, the expected payoff of C is at most $(1-\alpha) \cdot U^+ + \alpha \cdot \max\{U^-, U_{\text{random}}\}$. Let $U^-_{\text{rand}} = \max\{U^-, U_{\text{random}}\}$. By equation (1), it holds that $U \geq (k^d \cdot U^+ + U^-_{\text{rand}})/(k^d+1)$. Thus,

$$\begin{aligned}
(1-\alpha) \cdot U^+ + \alpha \cdot U^-_{\text{rand}} &\leq (1-\alpha) \cdot U^+ + \alpha \cdot U^-_{\text{rand}} + U - \frac{k^d \cdot U^+ + U^-_{\text{rand}}}{k^d+1} \\
&= U - \frac{\alpha k^d + \alpha - 1}{k^d+1}(U^+ - U^-_{\text{rand}}) \\
&\leq U - \frac{\alpha}{k^d+1}(U^+ - U^-_{\text{rand}}),
\end{aligned}$$

where the last inequality follows from the fact that $c \leq d$. From the assumption on utility functions, the difference $U^+ - U^-_{\text{rand}}$ is noticeable. Therefore, the expected payoff of C is noticeably less than U.

The rest of the proof, including the analysis for the deviations in the third or later rounds and a strict Nash equilibrium, is almost the same as the proof of Theorem 1. Therefore, we omit it. □

A protocol presented in [7] satisfies the property of S_3 in Theorem 2.

The condition (1) means that there is a polynomial factor gap between the amounts of payoffs when the player gains by learning the secret exclusively (i.e., $U^+ - U$) and when the player looses by learning the fake secret (i.e., $U - \max\{U^-, U_{\text{random}}\}$).

3.3 The Protocols with Immunity

We provide a general construction of a constant-round secret-sharing protocol that preserves both (strict) Nash equilibria and immunity of underlying protocols. To have the immunity, the protocol must proceed even if some players behave arbitrarily. At the same time, to achieve a strict Nash equilibrium, if some player deviated in the protocol, the payoff of the player must decrease.

The idea for achieving this goal is to have the protocol satisfy the property that if some player deviated in the protocol, the player cannot proceed to the later rounds. We implement it by symmetric-key encryption. If player i deviated, then in the later rounds the other players will broadcast their messages that are encrypted using symmetric-key encryption with a secret key player i does not have. Thereby, since the encrypted messages reveal no information to player i, player i is essentially excluded from the protocol. More concretely, the dealer generates a secret key sk_i^{SE} for each player i. The set of keys $\{sk_j^{\text{SE}}\}_{j \in N \setminus \{i\}}$ is included in the share of player i. If the other players detected a deviation of player i, they will encrypt messages by symmetric-key encryption with the key sk_i^{SE} in the later rounds.

In a strict Nash equilibrium, if some player deviated from the prescribed strategy, the payoff of the player must decrease by some noticeable amount. However, if we allow players to sample random strings, it is difficult to show that a subtle deviation from the protocol (e.g., sampling from a high-entropy distribution instead of a uniform one) decreases the payoff. As described above, we need secure symmetric-key encryption, which requires sampling random strings. To circumvent this problem, we use a pseudorandom function f for generating random strings. When player i deviated, the other players can use $f_s(r)$ as a random string at round r if the secret key s is not known to player i. This is because the string $f_s(r)$ is pseudorandom for players who do not know the secret key s. More concretely, the dealer generates secret keys $sk_{i,j}^{\mathrm{PRF}}$ for all $i, j \in N$ with $i \neq j$. The set of keys $\{sk_{i,j}^{\mathrm{PRF}}\}_{j \in N \setminus \{i\}}$ is included in all the shares of players $j \in N$ with $j \neq i$. If player i deviated in the protocol, player $j \neq i$ uses $f_{sk_{i,j}^{\mathrm{PRF}}}(r)$ for a random string at round r to encrypt a message. Since $f_{sk_{i,j}^{\mathrm{PRF}}}(r)$ is pseudorandom if $sk_{i,j}^{\mathrm{PRF}}$ is not known, messages are securely exchanged among the players $j \in N$ with $j \neq i$ without sampling random strings.

In the presence of an unexpectedly behaving player, if the shares of players have the standard n-out-of-n property that guarantees that any $n-1$ shares leak no information on the secret, then the players cannot reconstruct the secret if the unexpectedly behaving player does nothing in the protocol. Therefore, in the presence of t unexpectedly behaving players, we require the $(n-t)$-out-of-n property for a secret-sharing protocol instead of the n-out-of-n property.

We construct a constant-round protocol with 1-immunity based on any protocol with 1-immunity. Our protocol employs a symmetric-key encryption scheme, a family of pseudorandom functions $\mathcal{F} = \{f_{sk} \mid sk \in \{0,1\}^k\}_{k \in \mathbb{N}}$, an $(n-1)$-out-of-n authenticated secret-sharing scheme S_0, an $(\lfloor \frac{n}{2} \rfloor + 1)$-out-of-$n$ authenticated secret-sharing scheme S_1, an $\lceil \frac{n}{2} \rceil$-out-of-$n$ authenticated secret-sharing scheme S_2, and an $(n-1)$-out-of-n rational secret-sharing protocol S_3.

Sharing phase

To share a secret $s \in \{0,1\}^\ell$, the dealer performs the following:

- Choose $\mu \in \{0,1\}^\ell$ uniformly at random, and generate shares $(w_1, \ldots, w_n)$ of S_0 with the secret μ.
- Set $s' = \begin{cases} \mu \oplus s & \text{with probability } 1-\alpha, \\ \mathsf{fake} & \text{with probability } \alpha, \end{cases}$ where $\mathsf{fake} \in \{0,1\}^\ell$ is chosen uniformly at random, and generate shares $(x_1, \ldots, x_n)$ of S_1 with the secret s'.
- Set $s'' = 1$ if $s' = \mu \oplus s$ in the previous step, and $s'' = 0$ otherwise, and generate shares $(y_1, \ldots, y_n)$ of S_2 with the secret s''.
- Generate shares $(z_1, \ldots, z_n)$ of S_3 with the secret s.
- Generate a secret key sk_i^{SE} of the symmetric-key encryption scheme for each $i \in N$.
- Choose $sk_{i,j}^{\mathrm{PRF}} \in \{0,1\}^k$ uniformly at random for all $i \in N$ and $j \in N \setminus \{i\}$, and set $\eta_i = (sk_i^{\mathrm{SE}}, \{sk_{i,j}^{\mathrm{PRF}}\}_{j \in N \setminus \{i\}})$ for $i \in N$.
- Send $(w_i, x_i, y_i, z_i, \{\eta_j\}_{j \in N \setminus \{i\}})$ to player $i \in N$.

Reconstruction phase

After all the players received the shares, the players perform the following:

- For all $i \in N$ (in any order), send w_i.
  - After all the players broadcasted their messages, set N^* to be the set of players $j \in N$ who sent the valid share.
  - If $|N^*| \geq n-1$, reconstruct μ from $(w_1, \ldots, w_n)$ and go to the next round. Otherwise, halt and output a random string in $\{0,1\}^\ell$.

- For all $i \in N$ (in any order), if $|N^*| = n$, send x_i. Otherwise, encrypt x_i by symmetric-key encryption with the key $sk_{i'}^{\mathrm{SE}}$ using $f_{sk_{i',i}^{\mathrm{PRF}}}(2)$ as random bits, where $i' \notin N^*$, and send the ciphertext.
  - After all the players broadcasted their messages, if $|N^*| \neq n$, then decrypt the received ciphertexts. Update N^* to be the set of players $j \in N^*$ who sent the valid share. If $|N^*| \geq \lfloor \frac{n}{2} \rfloor + 1$, reconstruct s' from $(x_1, \ldots, x_n)$. Otherwise, set s' to be a random string in $\{0,1\}^\ell$.
  - If $|N^*| \geq n-1$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- For all $i \in N^*$ (in any order), if $|N^*| = n$, send y_i. Otherwise, encrypt y_i by symmetric-key encryption with the key $sk_{i'}^{\mathrm{SE}}$ using $f_{sk_{i',i}^{\mathrm{PRF}}}(3)$ as a random string, where $i' \notin N^*$, and send the ciphertext. (Each player is allowed to take any action when $s'' = 1$.)
  - After all the players broadcasted their messages, if $|N^*| \neq n$, then decrypt the received ciphertexts. Update N^* to be the set of players $j \in N^*$ who sent the valid share. If $|N^*| \geq \lceil \frac{n}{2} \rceil$, reconstruct s'' from $(y_1, \ldots, y_n)$.
  - If $|N^*| \geq n-1$ and $s'' = 0$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- If $|N^*| = n$, perform the reconstruction protocol of S_3 by using z_i to reconstruct s. Otherwise, perform the reconstruction protocol of S_3 in which player $i' \notin N^*$ deviates before starting the protocol, and at each round r, exchange a message by encrypting with the secret key $sk_{i'}^{\mathrm{SE}}$ using $f_{sk_{i',i}^{\mathrm{PRF}}}(r)$ as a random string.

  Then, halt and output s.

Theorem 3. *For any $n \geq 5$, the above is an $(n-1)$-out-of-n secret-sharing protocol that induces an $(\lceil \frac{n}{2} \rceil - 2, 1)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium if S_3 induces an $(\lceil \frac{n}{2} \rceil - 2, 1)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium, respectively. The secret is reconstructed in three rounds with probability at least $1-k^{-c}$, and the expected number of rounds for reconstruction is at most $3+\tau \cdot k^{-c}$ for any constant c, where k is the security parameter and τ is the expected number of rounds for reconstruction in S_3.*

Proof: By almost the same argument as the proof of Theorem 1, we can show the correctness and the $(n-1)$-out-of-n property in the presence of a single unexpectedly behaving player

We prove that the protocol induces an $(\lceil \frac{n}{2} \rceil - 2, 1)$-robust computational Nash equilibrium if S_3 induces an $(\lceil \frac{n}{2} \rceil - 2, 1)$-robust computational Nash equilibrium. Let σ be the prescribed strategy of the protocol. To prove the second condition of the robustness, namely, 1-immunity, we need to show that for any $i \in N$, any PPT strategy σ'_i, and $j \in N \setminus \{i\}$, we have that $u_j(\sigma) \leq u_j(\sigma_{-i}, \sigma'_i) + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$. Since the protocol satisfies the correctness, $u_j(\sigma) = U$. It is not difficult to see that, since S_3 induces 1-immunity, the protocol does not halt even if player i takes any PPT strategy σ'_i in the protocol. Thus, we have that $u_j(\sigma_{-i}, \sigma'_i) = U$, which proves 1-immunity of the protocol.

To prove the first condition of the robustness, we show that when player i^* takes an arbitrary PPT strategy ρ_{i^*}, the payoff of any coalition $C \subseteq N$ of size at most $\lceil \frac{n}{2} \rceil - 2$ does not increase under the assumption that the players in $N \setminus (C \cup \{i^*\})$ follow the protocol. Namely, we show that for any PPT strategy ρ_{i^*} of player i^* and any PPT strategy σ'_C of the coalition C, we have that $u_C(\sigma_{N\setminus(C\cup\{i^*\})}, \sigma'_C, \rho_{i^*}) \leq u_C(\sigma_{-i^*}, \rho_{i^*}) + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$. Note that $i^* \notin C$ from the definition. Let $N^\star = N \setminus (C \cup \{i^*\})$. Without loss of generality, we assume that ρ_{i^*} and σ'_C are deterministic strategies. Since the protocol induces 1-immunity, the payoff $u_C(\sigma_{-i^*}, \rho_{i^*})$ is equal to either U or the payoff when all the players in N except player i^* learn the secret. Hence, $u_C(\sigma_{-i^*}, \rho_{i^*}) \geq U$ from the definition of the payoff function. We will evaluate the value of $u_C(\sigma_{N^\star}, \sigma'_C, \rho_{i^*})$. Since the protocol induces 1-immunity, the payoff $u_C(\sigma_{N^\star}, \sigma'_C, \rho_{i^*})$ differs from $u_C(\sigma_{-i^*}, \rho_{i^*})$ only when at least two players in $C \cup \{i^*\}$ deviate from the protocol in the strategy profile $(\sigma_{N^\star}, \sigma'_C, \rho_{i^*})$. Let i_1 and i_2 be the first and the second player in $C \cup \{i^*\}$ who deviates from the protocol. We consider the following four cases: (1) player i_2 deviates in the first round; (2) player i_2 does not deviate in the first round, but in the second round; (3) player i_2 does not deviate in the first two rounds, but in the third round; (4) player i_2 does not deviate in the first three rounds, but in the fourth or later rounds; We will show that the payoff $u_C(\sigma_{N^\star}, \sigma'_C, \rho_{i^*})$ is at most $u_C(\sigma_{-i^*}, \rho_{i^*})$ in any case. Let $U'_C = u_C(\sigma_{N^\star}, \sigma'_C, \rho_{i^*})$, and $U_C = u_C(\sigma_{-i^*}, \rho_{i^*})$.

In case (1), the players in $N^\star$ do not proceed to the second or later rounds. The shares $\{w_i\}_{i\in N}$ only contain the information on μ. Also, even if the coalition C could obtain the share of player i^*, the shares $\{(x_i, y_i, z_i)\}_{i\in C\cup\{i^*\}}$ reveal no information on s' or s since the thresholds of S_1, S_2, and S_3 are strictly greater than $\lceil \frac{n}{2} \rceil - 1 \geq |C \cup \{i^*\}|$. Thus, the coalition C cannot learn the secret s, and the payoff of C is at most $\max\{U^-, U_{\mathrm{random}}\}$, which is noticeably less than U.

In case (2), the players in $N^\star$ do not proceed to the third or later rounds. However, every player in N (or $N \setminus \{i_1\}$ if player i_1 has deviated in the first round) can reconstruct s' since the players in $N^\star$ reveal their valid shares, and thus the number of valid shares revealed is at least $|N^\star| = n - (|C| + 1) \geq n - (\lceil \frac{n}{2} \rceil - 1) = \lfloor \frac{n}{2} \rfloor + 1$, which is at least the threshold of S_1. Even if the coalition C could obtain the share of player i^*, C cannot learn the secret s from $\{z_i\}_{i\in C\cup\{i^*\}}$. Therefore, since $s' = \mathsf{fake}$ with probability α, the expected payoff of C is at most $(1-\alpha) \cdot U_C + \alpha \cdot \max\{U^-, U_{\mathrm{random}}\}$, which is noticeably less than U_C.

In case (3), the players in $N^\star$ do not proceed to the fourth or later rounds. The fact that player i_2 deviated in the third round implies that $s' = \mathsf{fake}$. Then the coalition C cannot learn s since the only way to learn s is to use $\{z_i\}_{i\in C\cup\{i^*\}}$, but they are the shares of the $(n-1)$-out-of-n secret sharing S_3. Therefore, the payoff of C is at most $\max\{U^-, U_{\mathrm{random}}\}$, which is noticeably less than U.

In case (4), the players in N (or $N \setminus \{i_1\}$ if player i_1 has deviated in the first three rounds) proceed to the fourth or later rounds, which is the reconstruction protocol of S_3. Since S_3 induces an $(\lceil \frac{n}{2} \rceil - 2)$-resilient computational Nash equilibrium, the expected payoff of C is at most $U_C + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$.

In any case, we have shown that the expected payoff of C is $u_C(\sigma_{N \setminus (C \cup \{i^*\})}, \sigma'_C, \rho_{i^*}) \leq u_C(\sigma_{-i^*}, \rho_{i^*}) + \epsilon(k)$ for a negligible function $\epsilon(\cdot)$. Thus the protocol induces an $(\lceil \frac{n}{2} \rceil - 2, 1)$-robust computational Nash equilibrium.

Next we prove that the protocol induces a 1-resilient computational strict Nash equilibrium if S_3 induces a 1-resilient computational strict Nash equilibrium. Since we have shown in the above that the protocol induces a 1-resilient computational Nash equilibrium, we need to show that for any strategy σ'_i of player i such that $\sigma'_i \notin_{\mathsf{eq}} \sigma$, $u_i(\sigma'_i, \sigma_{-i}) \leq u_i(\sigma) - k^{-c}$ for a constant c. If player i deviates in the first round, although she may be able to reconstruct μ, she cannot understand the messages broadcasted by the other players in the later rounds since they are encrypted by symmetric-key encryption with a key she does not know. Since the other $n-1$ players can reconstruct the secret s, the payoff $u_i(\sigma'_i, \sigma_{-i})$ is at most $\max\{U^-, U_{\text{random}}\}$, which is noticeably less than U. If player i deviated in the second round, although she can reconstruct s', she cannot understand the messages broadcasted by the other players in the later rounds. Since s' is fake with probability α, the payoff $u_i(\sigma'_i, \sigma_{-i})$ is at most $(1-\alpha) \cdot U + \alpha \cdot \max\{U^-, U_{\text{random}}\}$, which is noticeably less than U. If player i deviated in the third round, which implies that s' is fake, player i need to participate in the reconstruction protocol of S_3 to reconstruct s. However, player i cannot understand the messages exchanged in the fourth or later rounds since they are encrypted by symmetric-key encryption. Thus, the payoff $u_i(\sigma'_i, \sigma_{-i})$ is at most $\max\{U^-, U_{\text{random}}\}$, which is noticeably less than U. If player i deviated in the fourth or later rounds, since S_3 induces a 1-resilient computational strict Nash equilibrium, the payoff $u_i(\sigma'_i, \sigma_{-i})$ is noticeably less than U. In any case, the deviation of player i decreases her payoff by a noticeable amount. This implies that the protocol induces a 1-resilient computational strict Nash equilibrium. □

We can extend the construction of Theorem 3 to preserve higher immunity. Specifically, we provide a construction of an $(n-t)$-out-of-n secret-sharing protocol that preserves an $(\lceil \frac{n}{2} \rceil - t - 1, t)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium, where t is any constant independent of k such that $1 \leq t \leq \lceil \frac{n}{2} \rceil - 1$. See Appendix B.3 for the description of the protocol.

We also provide a protocol that satisfies the property of S_3 in the above protocols. The protocol is a variant of the protocol given in [7], and is constructed based on the same idea of using symmetric-key encryption and pseudorandom functions. See Appendix B.4 for the description of the protocol and the proof of the property.

Optimality of Coalition Resilience

We discuss the optimality of our protocols with immunity regarding the coalition resilience.

First, we show that an (r,t)-robust protocol must have an $(r+t)$-coalition resilience.

Theorem 4. *If a secret-sharing protocol* Π *induces an* (r,t)*-robust computational Nash equilibrium, then* Π *induces an* $(r+t)$*-resilient computational Nash equilibrium.*

Proof: Let σ be the prescribed strategy of Π that induces an (r,t)-robust computational Nash equilibrium. Let C be a coalition with $|C| \leq r$, and $T \subset N \setminus C$ a set of players who behaves unexpectedly with $|T| \leq t$. It follows from the first condition of the (r,t)-robustness that $u_C(\sigma_{N\setminus(C\cup T)}, \sigma'_C, \rho_T) \leq u_C(\sigma_{-T}, \rho_T) + \epsilon(k)$ for any PPT strategies σ'_C and ρ_T, where σ is the prescribed strategy of Π and $\epsilon(\cdot)$ is a negligible function. The second condition of the (r,t) robustness implies that $u_C(\sigma_{-T}, \rho_T) \leq u_C(\sigma) + \epsilon'(k)$, where $\epsilon'(\cdot)$ is a negligible function. From these relations, we have that for any PPT strategy $\sigma'_{C\cup T}$, $u_C(\sigma_{N\setminus(C\cup T)}, \sigma'_{C\cup T}) \leq u_C(\sigma) + \delta(k)$ for a negligible function $\delta(\cdot)$. Then it follows from the definition of the payoff function for coalitions that $u_{C\cup T}(\sigma_{N\setminus(C\cup T)}, \sigma'_{C\cup T}) \leq u_{C\cup T}(\sigma) + \delta(k)$, which implies that Π induces an $(r+t)$-resilient computational Nash equilibrium. □

The next corollary immediately follows from the above theorem and the impossibility result of an $\lceil \frac{n}{2} \rceil$-resilient computational Nash equilibrium with constant-round reconstruction [2].

Corollary 1. *If a secret-sharing protocol in which the expected number of rounds for reconstruction is a constant (independent of the payoff of players) induces an (r,t)-robust computational Nash equilibrium, then $r + t \leq \lceil \frac{n}{2} \rceil - 1$.*

The above corollary implies that for an (r,t)-robust protocol with constant-round reconstruction, the coalition resilience $r = \lceil \frac{n}{2} \rceil - t - 1$ is optimal.

Let Π^* be the protocol presented in Section B.3, which is an extension of the protocol of Theorem 3. Since Π^* achieves an $(\lceil \frac{n}{2} \rceil - t - 1, t)$-robust computational Nash equilibrium for a constant t, the coalition resilience of Π^* is optimal among protocols that achieve t-immunity. Note that a construction of protocols for t that depends on k remains open.

Next we show that it is difficult to achieve strict Nash equilibrium and high immunity simultaneously.

Theorem 5. *Let Π be a secret sharing protocol for $n \geq 3$ players. If Π induces an (r,t)-robust computational strict Nash equilibrium with $r \geq 1$, then $t = 0$. If Π induces an r-resilient computational strict Nash equilibrium and computational 1-immunity, then $r \leq 1$.*

Proof: Assume for the contradiction that the prescribed strategy σ of Π is 1-immune. Suppose that player 1 takes any strategy σ'_1 such that $\sigma'_1 \not\in_{\mathsf{eq}} \sigma$. Since σ is 1-immune, the payoff of player 2 when the players take the strategy (σ'_1, σ_{-1}) is $u_2(\sigma'_1, \sigma_{-1}) \geq u_2(\sigma) - \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function, which implies that player 2 can reconstruct the secret with probability at least $1 - \epsilon'(k)$ for a negligible function $\epsilon'(\cdot)$. Consider the strategy ρ_2 of player 2 such that player 2 follows σ_2, and if the secret is reconstructed, then she broadcasts the secret. When the players follow the strategy $(\sigma'_1, \rho_2, \sigma_{-\{1,2\}})$, since player 1 can learn the secret with the same probability that player 2 can learn with, the payoff of player 1, namely, $u_1(\sigma'_1, \rho_2, \sigma_{-\{1,2\}})$, is at least $U - \epsilon''(k)$ for a negligible function $\epsilon''(\cdot)$. Thus, $u_1(\sigma'_1, \rho_2, \sigma_{-\{1,2\}}) \geq u_1(\sigma) - \epsilon''(k)$, which implies that σ does not satisfy the first condition of $(1,1)$-robust strict Nash equilibrium. Hence, the first statement follows. Furthermore, for the coalition $C = \{1,2\}$ we have $u_C(\sigma'_1, \rho_2, \sigma_{-C}) \geq u_C(\sigma) - \epsilon''(k)$, which follows from the definition of payoff functions of coalitions. Since $(\sigma'_1, \rho_2) \not\in_{\mathsf{eq}} \sigma$, this implies that σ does not induce a 2-resilient strict Nash equilibrium. Thus, the second statement follows. □

The first statement of Theorem 5 asserts that Π cannot achieve a computational strict Nash equilibrium *in the presence of* a malicious player. The second statement of Theorem 5 asserts that if Π achieves immunity, then the coalition resilience of strict Nash equilibrium must be at most 1.

Since Π^* induces t-immunity for a constant $t \geq 1$, it follows from Theorem 5 that 1-resilient computational strict Nash equilibrium is the maximum coalition resilience that we can hope for Π^*.

Acknowledgments

This study was supported in part by JSPS Global COE Program "Computationism as a Foundation for the Sciences," and JSPS Grant-in-Aid for Scientific Research Numbers 23500010, 23700010, 24240001, and 25106509.

A Strict Nash Equilibrium Implies Stability With Respect to Trembles

We show that a strict Nash equilibrium implies a Nash equilibrium that is stable with respect to trembles. This means that strict Nash equilibrium is a strong solution concept that captures stability against any small deviation of other players.

First, we define notions of the stability with respect to trembles, which was introduced in [7]. Intuitively, the stability with respect to trembles guarantees that even if a player believes that other players might follow any strategy other than the prescribed one with small probability, there is no better strategy for the player than the prescribed one. We say a PPT strategy profile ρ_{-i} is *δ-close* to σ_{-i} if ρ_{-i} takes σ_{-i} with probability $1-\delta$ and an arbitrary PPT strategy σ'_{-i} with probability δ.

Definition 7 (Stability with respect to trembles)**.** *A PPT strategy profile σ is a* computational Nash equilibrium that is stable with respect to trembles *if*

1. *σ is a computational Nash equilibrium;*
2. *There is a noticeable function $\delta(\cdot)$ such that for any $i \in N$, any PPT strategy profile ρ_{-i} that is δ-close to σ_{-i}, and any PPT strategy ρ_i, there exists a PPT strategy $\sigma'_i \in_{\mathsf{eq}} \sigma$ such that $u_i(\rho_i, \rho_{-i}) \leq u_i(\sigma'_i, \rho_{-i}) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function.*

Definition 8. *A PPT strategy profile σ is an* r-resilient computational Nash equilibrium that is stable with respect to trembles *if*

1. *σ is an r-resilient computational Nash equilibrium;*
2. *There is a noticeable function $\delta(\cdot)$ such that for any $C \subset N$ with $|C| \leq r$, any PPT strategy profile ρ_{-C} that is δ-close to σ_{-C}, and any PPT strategy ρ_C, there exists a PPT strategy $\sigma'_C \subset_{\mathsf{eq}} \sigma$ such that $u_C(\rho_C, \rho_{-C}) \leq u_C(\sigma'_C, \rho_{-C}) + \epsilon(k)$, where $\epsilon(\cdot)$ is a negligible function.*

We now prove that a strict Nash equilibrium implies the stability with respect to trembles.

Theorem 6. *If a secret-sharing protocol induces an r-resilient computational strict Nash equilibrium, then it also induces an r-resilient computational Nash equilibrium that is stable with respect to trembles.*

Proof: Let σ be a prescribed strategy of a secret-sharing protocol that induces an r-resilient computational strict Nash equilibrium. Let $C \subset N$ be any coalition with $|C| \leq r$ and ρ_{-C} any PPT strategy for players in $N \setminus C$ that is δ-close to σ_{-C} for some noticeable function $\delta(\cdot)$. We assume that ρ_{-C} takes σ_{-C} with probability $1-\delta$ and $\hat{\rho}_{-C}$ with probability δ. Let ρ_C be any PPT strategy for the players in C. We show that there exists a PPT strategy $\sigma'_C \subset_{\mathsf{eq}} \sigma$ such that

$U_C(\rho_C, \rho_{-C}) \leq U_C(\sigma'_C, \rho_{-C}) + \epsilon(k)$ for some negligible function $\epsilon(\cdot)$. Specifically, we show it by letting $\sigma'_C = \sigma_C$.

When $\rho_{-C} = \sigma_{-C}$, which occurs with probability $1 - \delta$, since σ is an r-resilient computational strict Nash equilibrium, we have

$$u_C(\rho_C, \rho_{-C}) - u_C(\sigma'_C, \rho_{-C}) = u_C(\rho_C, \sigma_{-C}) - u_C(\sigma_C, \sigma_{-C}) \leq -k^{-c_1},$$

where c_1 is some constant. When $\rho_{-C} = \hat{\rho}_{-C}$, which occurs with probability δ, the maximum payoff C can increase by changing the strategy from σ_C to ρ_C is at most $U^+ - U^-$. Thus,

$$\begin{aligned}
&u_C(\rho_C, \rho_{-C}) - u_C(\sigma_C, \rho_{-C}) \\
&\quad = (1-\delta)(u_C(\rho_C, \sigma_{-C}) - u_C(\sigma_C, \sigma_{-C})) + \delta(u_C(\rho_C, \hat{\rho}_{-C}) - u_C(\sigma_C, \hat{\rho}_{-C})) \\
&\quad \leq -k^{c_1} + \delta(U^+ - U^-) \\
&\quad \leq 0.
\end{aligned}$$

The last inequality follows if we take $\delta = k^{-c_3}$ for sufficiently large c_3. Therefore, the statement follows. □

B Other Protocols

B.1 The t-out-of-n Protocol

A general t-out-of-n protocol for $3 \leq t \leq n$ is constructed as a simple generalization of the n-out-of-n protocol. We employ a t-out-of-n authenticated secret-sharing scheme S_0, a $(\lfloor \frac{t}{2} \rfloor + 1)$-out-of-$n$ authenticated secret-sharing scheme S_1, a $\lceil \frac{t}{2} \rceil$-out-of-$n$ authenticated secret-sharing scheme S_2, and a t-out-of-n rational secret-sharing protocol S_3. The resulting protocol is an "exactly" t-out-of-n secret-sharing protocol, which works under the assumption that exactly t players exist in the reconstruction phase. We also assume that the coalition is a subset of the players in the reconstruction phase.

Since the sharing phase protocol is the same as the n-out-of-n case, we describe the reconstruction phase protocol.

Reconstruction phase

Let $M \subseteq N$ be the set of players in the reconstruction, where $|M| = t$. The players perform the following:

- For all $i \in N$ (in any order), send w_i.

  After all the players broadcasted their messages, if all the shares are valid, reconstruct μ from $\{w_j\}_{j \in M}$ and go to the next round. Otherwise, halt and output a random string in $\{0,1\}^\ell$.

- For all $i \in M$ (in any order), send x_i.

  - After all the players broadcasted their messages, set M^* to be the set of players $j \in M$ who sent the valid share. If $|M^*| \geq \lfloor \frac{t}{2} \rfloor + 1$, reconstruct s' from $\{x_i\}_{j \in M^*}$. Otherwise, set s' to be a random string in $\{0,1\}^\ell$.

– If $|M^*| = t$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- For all $i \in M$ (in any order), send y_i. (Each player is allowed to take any action when $s'' = 1$.)
  - After all the players broadcasted their messages, update M^* to be the set of players $j \in M^*$ who sent the valid share. If $|M^*| \geq \lceil \frac{t}{2} \rceil$, then reconstruct s'' from $\{y_i\}_{j \in M^*}$.
  - If $|M^*| = t$ and $s'' = 0$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.
- Perform the reconstruction protocol of S_3 by using z_i to reconstruct s.

  Then, halt and output s.

Theorem 7. *For any $n \geq 3$, the above is an exactly t-out-of-n secret-sharing protocol that induces a $(\lceil \frac{t}{2} \rceil - 1)$-resilient computational strict Nash equilibrium if S_3 induces a $(\lceil \frac{t}{2} \rceil - 1)$-resilient computational strict Nash equilibrium. The secret is reconstructed in three rounds with probability at least $1 - k^{-c}$, and the expected number of rounds for reconstruction is at most $3 + \tau \cdot k^{-c}$ for any constant c, where k is the security parameter and τ is the expected number of rounds for reconstruction in S_3.*

The proof is quite similar to that of Theorem 1.

We can also provide a general t-out-of-n protocol based on the same idea of [7]. Let $\Pi_{t,n}$ denote an exactly t-out-of-n protocol. In the general t-out-of-n protocol, the dealer prepares the shares of $\Pi_{t,n}, \Pi_{t+1,n}, \ldots, \Pi_{n,n}$. Then, in the reconstruction, the players perform the reconstruction protocol of $\Pi_{t^*,n}$ if there are t^* players in the reconstruction. It follows from Theorem 7 that the resulting t-out-of-n protocol achieves a $(\lceil \frac{t}{2} \rceil - 1)$-resilient computational strict Nash equilibrium.

B.2 The Protocol for Simultaneous Channels

If we assume a simultaneous broadcast channel instead of non-simultaneous broadcast channels, we can construct a simpler secret-sharing protocol. In simultaneous broadcast channels, in each round, every player can send one message and receive other players' messages, but the player must send a message before receiving other players' messages.

In the protocol for a non-simultaneous channel, players reconstruct s' in the second round and check the truthfulness of s' by reconstructing s'' in the third round. If the players use a simultaneous channel, we can achieve a strict Nash equilibrium even if players reconstruct s' and check the truthfulness of s' simultaneously. Thus, we need not to share the secret s'', and can set s' to be the special symbol $\perp$ if it is fake. We present an n-out-of-n protocol for simultaneous broadcast channels. Our protocol employs an n-out-of-n authenticated secret-sharing scheme S_0, an $\lceil \frac{n}{2} \rceil$-out-of-$n$ authenticated secret-sharing scheme S_1, and an n-out-of-n rational secret-sharing scheme S_2.

Sharing phase

To share a secret $s \in \{0,1\}^\ell$, the dealer performs the following:

- Choose $\mu \in \{0,1\}^\ell$ uniformly at random, and generate shares $(x_1, \ldots, x_n)$ of S_0 with the secret μ.

- Set $s' = \begin{cases} s & \text{with probability } 1-\alpha, \\ \perp & \text{with probability } \alpha, \end{cases}$ and generate shares $(y_1, \ldots, y_n)$ of S_1 with the secret s'.

- Generate shares $(z_1, \ldots, z_n)$ of S_2 with the secret s.

- Send (x_i, y_i, z_i) to player $i \in N$.

Reconstruction phase

After all the players received the shares, player i performs the following:

- Send x_i.

  If all the received shares are valid, reconstruct μ from $(x_1, \ldots, x_n)$ and go to the next round. Otherwise, halt and output a random string in $\{0,1\}^\ell$.

- Send y_i.

  - Set N^* to be the set of players $j \in N$ who sent the valid share. If $|N^*| \geq \lceil \frac{n}{2} \rceil$, reconstruct s' from $(y_1, \ldots, y_n)$. Otherwise, set s' to be a random string in $\{0,1\}^\ell$.
  - If $|N^*| = n$ and $s' = \perp$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- Perform the reconstruction protocol of S_2 by using z_i to reconstruct s.

  Then, halt and output s.

Theorem 8. *For any $n \geq 3$, the above is an n-out-of-n secret-sharing protocol that induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium in a simultaneous broadcast channel if S_2 induces an $(\lceil \frac{n}{2} \rceil - 1)$-resilient computational strict Nash equilibrium. The secret is reconstructed in two rounds with probability at least $1 - k^{-c}$, and the expected number of rounds for reconstruction is at most $2 + \tau \cdot k^{-c}$ for any constant c, where k is the security parameter and τ is the expected number of rounds for reconstruction in S_2.*

The proof is similar to that of Theorem 1.

B.3 The Protocol with Higher Immunity

We present a constant-round protocol with t-immunity based on any protocol with t-immunity for any constant $t \geq 1$ that is independent of the security parameter k. The idea is a simple generalization of the 1-immune protocol presented in Section 3.3. If some set T of players with $|T| \leq t$ deviated in the protocol, then in the later rounds, the other players will broadcast their messages that are encrypted using symmetric-key encryption with a secret key that the players in T do not have. To implement this idea, we prepare 2^t keys for the deviations of any set of at most t players. Therefore, this protocol works if t is a constant independent of k. Our protocol employs a symmetric-key encryption scheme, a family of pseudorandom functions $\mathcal{F} = \{f_{sk} \mid sk \in \{0,1\}^k\}_{k \in \mathbb{N}}$, an $(n-t)$-out-of-n authenticated secret-sharing scheme S_0, an $(\lfloor \frac{n}{2} \rfloor + 1)$-out-of-$n$ authenticated secret-sharing scheme S_1, an $\lceil \frac{n}{2} \rceil$-out-of-$n$ authenticated secret-sharing scheme S_2, and an $(n-t)$-out-of-n rational secret-sharing protocol S_3.

Sharing phase

To share a secret $s \in \{0,1\}^\ell$, the dealer performs the following:

- Choose $\mu \in \{0,1\}^\ell$ uniformly at random, and generate shares $(w_1, \ldots, w_n)$ of S_0 with the secret μ.
- Set $s' = \begin{cases} \mu \oplus s & \text{with probability } 1-\alpha, \\ \mathsf{fake} & \text{with probability } \alpha, \end{cases}$, where $\mathsf{fake} \in \{0,1\}^\ell$ is chosen uniformly at random, and generate shares $(x_1, \ldots, x_n)$ of S_1 with the secret s'.
- Set $s'' = 1$ if $s' = \mu \oplus s$ in the previous step, and $s'' = 0$ otherwise, and generate shares $(y_1, \ldots, y_n)$ of S_2 with the secret s''.
- Generate shares $(z_1, \ldots, z_n)$ of S_3 with the secret s.
- Generate a secret key sk_T^{SE} of the symmetric-key encryption scheme for each $T \in P_t(N)$, where $P_t(N)$ is the family of all subsets of N of size at most t.
- Choose $sk_{T,j}^{\mathrm{PRF}} \in \{0,1\}^k$ uniformly at random for all $T \in P_t(N)$ and $j \in N \setminus T$, and set $\eta_T = (sk_T^{\mathrm{SE}}, \{sk_{T,j}^{\mathrm{PRF}}\}_{j \in N \setminus T})$ for $T \in P_t(N)$.
- Send $(x_i, y_i, z_i, \{\eta_T\}_{T \in P_t(N), i \notin T})$ to player $i \in N$.

Reconstruction phase

After all the players received the shares, the players perform the following:

- For all $i \in N$ (in any order), send w_i.
  - After all the players broadcasted their messages, set N^* to be the set of players $j \in N$ who sent the valid share, and set $T^* = N \setminus N^*$.
  - If $|N^*| \geq n - t$, reconstruct μ from $(w_1, \ldots, w_n)$ and go to the next round. Otherwise, halt and output a random string in $\{0,1\}^\ell$.
- For all $i \in N^*$ (in any order), if $|N^*| = n$, send x_i. Otherwise, encrypt x_i by symmetric-key encryption with the key $sk_{T^*}^{\mathrm{SE}}$ using $f_{sk_{T^*,i}^{\mathrm{PRF}}}(2)$ as random bits, and send the ciphertext.
  - After all the players broadcasted their messages, if $|N^*| \neq n$, then decrypt the received ciphertexts. Update N^* to be the set of players $j \in N^*$ who sent the valid share, and T^* to be $N \setminus N^*$. If $|N^*| \geq \lfloor \frac{n}{2} \rfloor + 1$, reconstruct s' from $(x_1, \ldots, x_n)$. Otherwise, set s' to be a random string in $\{0,1\}^\ell$.
  - If $|N^*| \geq n - t$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.
- For all $i \in N^*$ (in any order), if $|N^*| = n$, send y_i. Otherwise, encrypt y_i by symmetric-key encryption with the key $sk_{T^*}^{\mathrm{SE}}$ using $f_{sk_{T^*,i}^{\mathrm{PRF}}}(3)$ as a random string, and send the ciphertext. (Each player is allowed to take any action when $s'' = 1$.)

– After all the players broadcasted their messages, if $|N^*| \neq n$, then decrypt the received ciphertexts. Update N^* to be the set of players $j \in N^*$ who sent the valid share, and T^* to be $N \setminus N^*$. If $|N^*| \geq \lceil \frac{n}{2} \rceil$, then reconstruct s'' from $(y_1, \ldots, y_n)$.

– If $|N^*| \geq n - t$ and $s'' = 0$, go to the next round. Otherwise, halt and output $s' \oplus \mu$.

- If $|N^*| = n$, perform the reconstruction protocol of S_3 by using z_i to reconstruct s. Otherwise, perform the reconstruction protocol of S_3 in which the players in T^* deviate before starting the protocol, and at each round r, exchange a message by encrypting with the secret key $sk_{T^*}^{\text{SE}}$ using $f_{sk_{T^*,i}^{\text{PRF}}}(r)$ as a random string.

  Then, halt and output s.

Theorem 9. *For any $n \geq 3$ and any constant $t \geq 1$ (independent of k), the above is an $(n-t)$-out-of-n secret-sharing protocol that induces an $(\lceil \frac{n}{2} \rceil - t - 1, t)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium if S_3 induces an $(\lceil \frac{n}{2} \rceil - t - 1, t)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium, respectively. The secret is reconstructed in three rounds with probability at least $1 - k^{-c}$, and the expected number of rounds for reconstruction is at most $3 + \tau \cdot k^{-c}$ for any constant c, where k is the security parameter and τ is the expected number of rounds for reconstruction in S_3.*

The proof is similar to that of Theorem 3.

B.4 The Protocol with Immunity Based on the Protocol of [7]

We present a protocol that satisfies the property of S_3 in Theorems 3 and 9. The protocol is based on the protocol of [7]. The idea for achieving immunity is almost the same as the protocols presented in Sections 3.3 and B.3. The protocol uses as building blocks two verifiable random functions $(\mathsf{GenVRF}, \mathsf{Eval}, \mathsf{Prove}, \mathsf{VerVRF})$ and $(\mathsf{GenVRF}', \mathsf{Eval}', \mathsf{Prove}', \mathsf{VerVRF}')$, a symmetric encryption scheme, and a family of pseudorandom functions $\mathcal{F} = \{f_{sk} \mid sk \in \{0,1\}^k\}_{k \in \mathbb{N}}$. The definition of verifiable random function is provided in Appendix C.2.

Sharing phase

To share a secret $s \in \{0,1\}^\ell$, the dealer performs the following:

- Generate a secret key sk_T^{SE} of the symmetric encryption scheme for each $T \in P_t(N)$.
- Choose $sk_{T,j}^{\text{PRF}} \in \{0,1\}^k$ uniformly at random for all $T \in P_t(N)$ and $j \in N \setminus T$, and set $\eta_T = (sk_T^{\text{SE}}, \{sk_{T,j}^{\text{PRF}}\}_{j \in N \setminus T})$ for $T \in P_t(N)$.
- Choose $r^* \in \mathbb{N}$ according to a geometric distribution with parameter β.
- Generate $(pk_i, sk_i) \leftarrow \mathsf{GenVRF}(1^k)$ and $(pk_i', sk_i') \leftarrow \mathsf{GenVRF}'(1^k)$ for $i \in N$.
- Choose random polynomials G and H of degree $n - t - 1$ such that $G(0) = s$ and $H(0) = 0$, where $G(i) \in \{0,1\}^\ell$ and $H(i) \in \{0,1\}^k$ for $i \in N$.
- Send $(\{\eta_T\}_{T \in P_t(N), i \notin T}, sk_i, sk_i')$ to player $i \in N$, and the following to all players:

- $\{(pk_i, pk'_i)\}_{i \in N}$
- $\{g_i = G(i) \oplus \mathsf{Eval}_{sk_i}(r^*)\}_{i \in N}$
- $\{h_i = H(i) \oplus \mathsf{Eval}'_{sk'_i}(r^* + 1)\}_{i \in N}$

Reconstruction phase

After all the players received the shares, set $N^* = N$ and $T^* = \emptyset$. Each player i chooses $s_i^{(0)} \in \{0,1\}^\ell$ uniformly at random. In each round $r = 1, \ldots,$ player i performs the following:

- Compute
$$v_i^{(r)} = (\pi_i^{(r)}, \rho_i^{(r)}, \mathsf{Prove}_{sk_i}(r), \mathsf{Prove}'_{sk'_i}(r)),$$
where $\pi_i^{(r)} = \mathsf{Eval}_{sk_i}(r)$ and $\rho_i^{(r)} = \mathsf{Eval}'_{sk'_i}(r)$. If $|N^*| = n$, send $v_i^{(r)}$. Otherwise, encrypt $v_i^{(r)}$ by symmetric encryption with the key $sk_{T^*}^{\mathrm{SE}}$ using $f_{sk_{T^*,i}^{\mathrm{PRF}}}(r)$ as a random string. Then, send the ciphertext. (Each player is allowed to take any action when $H^{(r)}(0) = 0$, where $H^{(r)}$ is defined below.)
- After all the players broadcasted their messages, if $|N^*| \neq n$, then decrypt the received ciphertexts. Update N^* to be the set of players $j \in N^*$ who sent the correct proof, and T^* to be $N \setminus N^*$.
  - If $|N^*| < n-t-1$, then halt and output $s_i^{(r-1)}$. Otherwise, set $h_j^{(r)} = h_j \oplus \rho_j^{(r)}$ for $j \in N^*$, and interpolate a polynomial $H^{(r)}$ of degree $n-t-1$ through the points $\{h_j^{(r)}\}_{j \in N^*}$.
  - If $H^{(r)}(0) = 0$, then halt and output $s_i^{(r-1)}$. Otherwise, set $g_j^{(r)} = g_j \oplus \pi_j^{(r)}$ for $j \in N^*$, interpolate a polynomial $G^{(r)}$ of degree $n-t-1$ through the points $\{g_j^{(r)}\}_{j \in N^*}$, and set $s_i^{(r)} = G^{(r)}(0)$.

Note that a variant of the "exactly" $(n-t-1)$-out-of-n protocol of [7] is used in the above protocol. The parameter β is chosen to be a sufficiently small value that depends on t and the payoff of players.

Theorem 10. *For any $n \geq 3$ and any constant $t \geq 1$ (independent of k and the payoff), the above is a secret-sharing protocol that induces an $(n-t-1,t)$-robust computational Nash equilibrium and a 1-resilient computational strict Nash equilibrium. The expected number of rounds for reconstruction is $O(\beta^{-1})$ where β is a sufficiently small value depending on t and the payoff of players.*

Proof: First we prove that the protocol induces an $(n-t-1,t)$-robust computational Nash equilibrium. Since the protocol does not halt as long as at most t players deviate, it satisfies the second condition of the robustness, namely, t-immunity. To prove the first condition, we show that even if players in T with $|T| \leq t$ take any strategy, the payoff of any coalition $C \subseteq N$ with $|C| \leq n-t-1$ does not increase under the assumption that the players in $N \setminus (C \cup T)$ follow the protocol. The case that the payoff of C is strictly larger than U is that C learns the secret, but players in $N \setminus C$ do not. This situation can be achieved only if the coalition C successfully predicts the "real" round r^*. The coalition has at most $t+1$ trials to predict r^* since the protocol induces

t-immunity. Since r^* is chosen according to a geometric distribution, the probability of being the real round is the same in any round before the real one. Therefore, the expected payoff of C greater than that of the protocol without immunity (namely, the original protocol of [7]) is at most $t \cdot U^+$. Hence, if we choose β to satisfy that $U > \beta \cdot (t+1) \cdot U^+ + (1-\beta) \cdot U^-$ (the condition on β in the protocol of [7] is the case $t = 0$), then the expected payoff of C does not increase. We can choose such β if t is a constant independent of the payoff.

Next we prove that the protocol induces a 1-resilient computational strict Nash equilibrium. If a player deviated, the player cannot understand the message exchanged in the later rounds since they are encrypted, and thus the player cannot learn the secret. Therefore, a single deviation decreases the payoff by a noticeable amount. □

C Cryptographic Primitives

C.1 Authenticated Secret-Sharing Scheme

An authenticated secret-sharing scheme is a secret-sharing scheme with authentication, which can be obtained by a standard technique [25, 22]. The dealer generates shares $(\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_n)$ of secret $s \in \{0,1\}^k$ by $\mathsf{GenSS}(s)$, where $\hat{\sigma}_i = (\sigma_i, \pi_i, v_i)$. Players can reconstruct the secret if a sufficient number of σ_i are collected. Each player i can verify whether a collected share (σ_j, π_j) is valid or not by using v_i.

Definition 9. *An* m-out-of-n authenticated secret-sharing scheme *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{GenSS}, \mathsf{Rec}, \mathsf{VerSS})$ *such that*

- *On input* $s \in \{0,1\}^\ell$ *and* 1^k, GenSS *outputs* $(\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_n)$, *where* $\hat{\sigma}_i = (\sigma_i, \pi_i, v_i)$ *and* ℓ *is a polynomial in* k.
- ***Correctness:*** *For any* $M \subset \{1, \ldots, n\}$ *with* $|M| \geq m$, $\mathsf{Rec}(\{\sigma_i\}_{i \in M}) = s$. *Also, for any* $i, j \in \{1, \ldots, n\}$, $\mathsf{VerSS}(v_i, (\sigma_j, \pi_j)) = 1$.
- ***Security:*** *For any* $s, s' \in \{0,1\}^\ell$, *and* $M \subset \{1, \ldots, n\}$ *with* $|M| < m$, *two sets of random variables* $\{\sigma_i\}_{i \in M}$ *and* $\{\sigma'_i\}_{i \in M}$ *are identically distributed, where* $(\hat{\sigma}_1, \ldots, \hat{\sigma}_n) \leftarrow \mathsf{GenSS}(s, 1^k)$ *and* $(\hat{\sigma}'_1, \ldots, \hat{\sigma}'_n) \leftarrow \mathsf{GenSS}(s', 1^k)$.
- ***Authenticity:*** *For any* $s \in \{0,1\}^\ell$, $i \in \{1, \ldots, n\}$, *and algorithm* A,

$$\Pr[\mathsf{VerSS}(v_i, (\sigma', \pi')) = 1 \wedge \sigma' \notin \{\sigma_j\}_{j \neq i}] \leq \epsilon(k)$$

  for every k, *where* $(\hat{\sigma}_1, \ldots, \hat{\sigma}_n) \leftarrow \mathsf{GenSS}(s, 1^k)$, $(\sigma', \pi') \leftarrow A(\{\hat{\sigma}_j\}_{j \neq i})$, *and* $\epsilon(\cdot)$ *is a negligible function.*

C.2 Verifiable Random Function

A verifiable random function is a pseudorandom function with verifiability, which was introduced by Micali, Rabin, and Vadhan [17]. We need a verifiable random function with unique proofs. The constructions for such stronger verifiable random functions were provided in [4] and [6].

Definition 10. *A* verifiable random function *is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{GenVRF}, \mathsf{Eval}, \mathsf{Prove}, \mathsf{VerVRF})$ *such that*

- *On input* 1^k*,* GenVRF *outputs* (pk, sk)*.*
- ***Correctness:*** *For any* $x \in \{0,1\}^k$*,* $\mathsf{VerVRF}_{pk}(x, \mathsf{Eval}_{sk}(x), \mathsf{Prove}_{sk}(x)) = 1$*.*
- ***Verifiability:*** *There does not exist a tuple* (x, y, π, y', π') *with* $y \neq y'$ *such that* $\mathsf{VerVRF}(x, y, \pi) = \mathsf{VerVRF}(x, y', \pi') = 1$*.*
- ***Unique proofs:*** *There does not exist a tuple* (x, y, π, π') *with* $\pi \neq \pi'$ *such that* $\mathsf{VerVRF}(x, y, \pi) = \mathsf{VerVRF}(x, y, \pi') = 1$*.*
- ***Pseudorandomness:*** Eval *is a pseudorandom function.*

References

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

[2] G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. *J. Cryptology*, 24(1):157–202, 2011.

[3] R. Canetti, editor. *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*. Springer, 2008.

[4] Y. Dodis. Efficient construction of (distributed) verifiable random functions. In Y. Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2003.

[5] Y. Dodis and T. Rabin. Cryptography and game theory. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algoritmic Game Theory*, pages 181–207. Cambridge University Press, New York, NY, USA, 2007.

[6] Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In S. Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431. Springer, 2005.

[7] G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In D. Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2010.

[8] J. A. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *FOCS*, pages 648–657. IEEE Computer Society, 2013.

[9] S. D. Gordon and J. Katz. Rational secret sharing, revisited. In R. D. Prisco and M. Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 229–241. Springer, 2006.

[10] J. Y. Halpern. computer science and game theory. In S. N. Durlauf and L. E. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke, 2008.

[11] J. Y. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In L. Babai, editor, *STOC*, pages 623–632. ACM, 2004.

[12] J. Katz. Bridging game theory and cryptography: Recent results and future directions. In Canetti [3], pages 251–272.

[13] G. Kol and M. Naor. Cryptography and game theory: Designing protocols for exchanging information. In Canetti [3], pages 320–339.

[14] G. Kol and M. Naor. Games for exchanging information. In C. Dwork, editor, *STOC*, pages 423–432. ACM, 2008.

[15] N. Linial. Game-theoretic aspects of computing. In R. Aumann and S. Hart, editors, *Handbook of Game Theory With Economic Applications*, volume 2, chapter 38, pages 1339–1395. North-Holland, 1994.

[16] A. Lysyanskaya and N. Triandopoulos. Rationality and adversarial behavior in multi-party computation. In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 180–197. Springer, 2006.

[17] S. Micali, M. O. Rabin, and S. P. Vadhan. Verifiable random functions. In *FOCS*, pages 120–130. IEEE Computer Society, 1999.

[18] S. Micali and A. Shelat. Purely rational secret sharing (extended abstract). In Reingold [23], pages 54–71.

[19] J. B. Nielsen, editor. *Summary Report on Rational Cryptographic Protocols*, ECRYPT Report, 2007.

[20] S. J. Ong, D. C. Parkes, A. Rosen, and S. P. Vadhan. Fairness with an honest minority and a rational majority. In Reingold [23], pages 36–53.

[21] R. Pass and A. Shelat. Renegotiation-safe protocols. In B. Chazelle, editor, *ICS*, pages 61–78. Tsinghua University Press, 2011.

[22] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In D. S. Johnson, editor, *STOC*, pages 73–85. ACM, 1989.

[23] O. Reingold, editor. *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*. Springer, 2009.

[24] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[25] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.