Games For Exchanging Information*

[Extended Abstract]

Gillat Kol
Weizmann Institute of Science
Rehovot Israel
gillat.kol@weizmann.ac.il

Moni Naor†
Weizmann Institute of Science
Rehovot Israel
moni.naor@weizmann.ac.il

ABSTRACT

We consider the *rational* versions of two of the classical problems in foundations of cryptography: secret sharing and multiparty computation, suggested by Halpern and Teague (STOC 2004). Our goal is to design games and fair strategies that encourage rational participants to exchange information about their inputs for their mutual benefit, when the only mean of communication is a broadcast channel.

We show that protocols for the above information exchanging tasks, where players' values come from a bounded domain, cannot satisfy some of the most desirable properties. In contrast, we provide a rational secret sharing scheme with simultaneous broadcast channel in which shares are taken from an unbounded domain, but have finite (and polynomial sized) expectation.

Previous schemes (mostly cryptographic) have required computational assumptions, making them inexact and susceptible to backward induction, or used stronger communication channels. Our scheme is non-cryptographic, immune to backward induction, and satisfies a stronger rationality concept (strict Nash equilibrium). We show that our solution can also be used to construct an ε-Nash equilibrium secret sharing scheme for the case of a *non*-simultaneous broadcast channel.

Categories and Subject Descriptors

F.m [**Theory of Computation**]: Miscellaneous

General Terms

Theory

Keywords

Game Theory, Cryptography, Secret Sharing, Multiparty Computation, Nash Equilibrium, Backward Induction

*Research supported by a grant from the Israel Science Foundation.
†Incumbent of the Judith Kleeman Professorial Chair.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.

1. INTRODUCTION

1.1 Background

We consider rational (in a Game Theoretic sense) versions of two classical cryptographic problems, Secret Sharing and Multiparty Computation (MPC), introduced by Halpern and Teague [5]. In the classical problem of m-out-of-n secret sharing, a dealer wishes to entrust a secret with a group of n players such that any subset of m or more players can reconstruct the secret, but a subset of less than m players cannot learn anything about the secret. In the problem of mutiparty computation, a group of players wish to evaluate a function on private inputs with no external help.[1] Note that secret sharing and MPC are closely connected: In a secret sharing scheme, players run an MPC protocol on their shares in order to reconstruct the secret.

The traditional cryptographic setting assumes that players are either arbitrarily malicious or totally honest. However, in some situations it may make more sense to view the players as rational individuals trying to maximize their own gain. The goal of this work is to design *fair*, *stable* protocols in such rational settings allowing all players to learn the designated value (the secret or the function's value). Of course, such a task is only possible if the players have an initial incentive to collaborate. As suggested in [5], to motivate players to cooperate, we assume that they prefer getting the value to not getting it. In some cases it is further assumed that players prefer to get the secret while others do not.

However, even if players have an initial incentive to collaborate, they will only follow the protocol if they cannot gain from deviating. The best known Game Theoretic concept capturing this "stability" demand is that of a Nash equilibrium: A protocol is a Nash equilibrium if no player can get a higher payoff by deviating from his prescribed strategy, given that all the others are following their strategies. In other words, in a Nash equilibrium, each player's strategy is a *best response* to the strategies of the others.

The main difficulty in designing such stable protocols is the players' tendency to deviate in the last round of the

[1]The MPC problem is easy (trivially) in the usual cryptographic setting, and in any model for which no party tries to prevent others from learning the function's value: Each party simply sends his share to the others. In our setting players prefer to learn the function's value alone, making rational MPC non-trivial.
The classical secure multiparty computation problem can be viewed as the task of finding an MPC protocol that reveals no additional information about the players' inputs, over what is already disclosed by the function.

protocol and keep their information to themselves. In order demonstrate the problem, recall the m-out-of-n secret sharing scheme due to Shamir [14]: The dealer chooses a random polynomial $p(x)$ of degree at most $m-1$ with a free coefficient that is the secret, and gives the share $p(i)$ to player i. Any set of m players can recover p (and hence the secret) by broadcasting their shares and interpolating the polynomial, while no set of fewer than m players can deduce any information about the secret.

Although Shamir's scheme allows m honest players to learn the secret, it fails to do so in our rational settings. Denote by t the number of players participating in the reconstruction. For $t = m$ Shamir's scheme is not a Nash equilibrium: Players will simply prefer to keep silent rather than broadcast their shares in order to learn the secret alone. When $t > m$, the scheme is an equilibrium, however players still prefer to keep silent, since the silence strategy is never worse than the broadcasting one, and it is sometimes strictly better (e.g., if for some reason exactly $m-1$ other players broadcast).

The above example suggests that the rational settings are at times more challenging than the standard ones. However, the two settings are really incomparable: Although *perfect* fairness cannot be achieved in the usual cryptographic settings (i.e., it is possible that one party obtains his desired output while others do not), it is possible in our Game Theoretic setting. The key difference is that the specified restrictions on players' preferences allow us to punish a deviating player by preventing him from learning the designated value, whereas in the usual cryptographic setting malicious players cannot be punished - they simply do not care about learning.

In this work we first discuss the desired Game Theoretic properties of protocols for exchanging information. We argue that the previous *iterated admissibility* (a.k.a, surviving iterated elimination of weakly dominated strategies) criterion used to evaluate such protocols is problematic, and suggest the stronger notion of *strict equilibrium*. Furthermore, we show that previously suggested protocols in similar settings are susceptible to *backward induction*. We propose the new notion of *everlasting equilibrium* that ensures immunity to backward induction.

Our main contributions are tight positive and negative results concerning MPC and secret sharing, when the communication between players is via a broadcast channel. We consider the case of a *simultaneous* broadcast channel (SBC), where all player broadcast messages at the same time (no rushing), as well as the case of a *non-simultaneous* broadcast channel (NSBC), where there is only a single sender per round. For the 2 players case we show that *no* function with a *bounded domain* can be computed using a Nash equilibrium protocol, even when the communication between the players is via an SBC. We then conclude that there is no Nash equilibrium, 2-out-of-n secret sharing scheme, assigning the players shares that are taken from a bounded domain. Our work holds for the NSBC model as well, and extends (with some restrictions) to rational MPC with any number of players, and to rational m-out-of-n secret sharing, for any $2 \leq m \leq n$.

In contrast, we show that by allowing (finite) shares taken from *unbounded domains* we can obtain an m-out-of-n secret sharing scheme that is a strict everlasting Nash equilibrium for the simultaneous model, and an ε-everlasting Nash equilibrium for the non-simultaneous model. The schemes are designed to be efficient, although both our possibility and impossibility results hold for players with unbounded computational resources as well. As far as we know, this is the first result connecting the ability to achieve good protocols to the boundedness of the domain, and also the first result regarding the non-simultaneous model[2].

1.2 Related Works

Several information exchange protocols were offered by Halpern and Teague [5], Gordon and Katz [4], Abraham et al. [1], and Lysyanskaya and Triandopoulos [10]. The key idea used is that in any given round, players do not know whether the current round is going to be the last round, or whether this is just a test round designed to catch cheaters. The protocol suggested in [1] is coalition-proof, and in [1, 10] the case of "mixed" security (when both arbitrarily malicious and rational players might be present) is considered.

All the above results assume simultaneous channels (either a broadcast channel or secure private channels). The protocols in [4, 1, 10] use cryptographic techniques relying on computational assumptions, and achieve approximated equilibria under the assumption that players can only run efficient strategies.

Another line of work was pursued by Lepinski et al. [8, 9] and Izmalkov et al. [6] in their recent sequence of papers. Roughly speaking, they were able to obtain fair, rational SMPC protocols, prevent coalitions, and eliminate subliminal channels. However, the hardware requirements needed for these operations, including ideal envelopes and ballot boxes, are very strict; it is not clear how they can be implemented for distant participants, if at all.

1.3 Our Contributions

The rest of this section lists our results and the organization of the paper:

Solution Concept (Section 2): When Game Theory and Cryptography are mixed, the "standard" CS intuition often turns out to be false, and delicate Game Theoretic considerations must also be taken into account. We point out two such problematic issues, and offer new solution concepts intended to correct them.

In Section 2.1 we argue that the iterated admissibility criterion suggested in [5] and adopted by [4, 1, 10], should not be used to distinguish "good" information exchange protocols from "bad" ones, since many bad strategies are not ruled out by it. Instead, we suggest the stronger notion of strict Nash equilibrium, in which every player's strategy is a *strict* best response. Due to the restrictive nature of this notion, we regard it as a sufficient condition and not as a necessary one. It is only used in the positive results, while the impossibility results use a much weaker criterion.

In Section 2.2 we claim that the previously suggested protocols for the SBC model, making use of cryptographic tools, are problematic: After an exponential number of rounds (say b rounds) the cryptographic primitives can be broken, thus players will no longer follow their strategies if round b is reached. Furthermore, using a the Game Theoretic backward induction process, it can be shown that players prefer

[2]There has been quite a lot of effort into approximating an SBC via an NSBC using cryptographic techniques and obtaining fair protocols (see [2, 3, 12] for recent work). Note, however, that such results do not take into account the rationality consideration that we use in this work.

to deviate from the start. To prevent such phenomenon from taking place, we suggest the notion of everlasting equilibrium that ensures that players' strategies are best responses after any sequence of rounds.

Impossibility Results (Section 4): We define the notion of a revelation point in a rational MPC protocol, and use it to rule out "unreasonable" protocols. A revelation point is a point in the execution of a protocol, recognizable by all the players, for which some players still do not know the value $f(\mathbf{x})$, however at any point after the revelation point, $f(\mathbf{x})$ is known to everyone. Informally speaking, protocols with revelation points are problematic from the following reason: We expect rational players not to broadcast any meaningful information when a revelation point is reached, since they learn $f(\mathbf{x})$ during the next round anyway. However, since everyone learns the value after the revelation point, some players must have given out information.

We show that in both the SBC and NSBC communication models, for *every* non-constant function f with a *finite domain* there is no Nash equilibrium protocol without a revelation point that computes f. We then deduce that there are no *strict* equilibria protocols for MPC of *any* non-constant function, and that there are no plain Nash equilibria protocols for two parties.

A Strict Rational Scheme with Unbounded Shares (Section 5): Since every secret sharing scheme requires the players to evaluate a non-trivial function of their shares, the impossibility results imply that there is no "reasonable" exact Nash equilibria, secret sharing schemes with shares taken from *finite* sets.

One way of getting a positive result is allowing (finite) shares taken from *infinite domains*. We present such a *strict* everlasting equilibrium scheme that uses an SBC. The key idea is to assign players shares of different lengths, and use the uncertainty of each player as to the lengths of the shares assigned to the others, to prevent players from foreseeing which iteration is last.

An ε-Rational Scheme for the NSBC Model (Section 6): For the NSBC model, no strict equilibria or even plain Nash equilibria protocols cannot be obtained, even when allowing unbounded shares (at least in the 2 players case). Therefore, we settle for the relaxed notion of ε-Nash equilibrium: An ε-Nash equilibrium protocol is close to equilibrium in the sense that no player can gain more than ε by deviating.

In this section we offer such ε-Nash equilibrium secret sharing scheme for the case of an NSBC, where ε is exponentially small in the share sizes. The scheme is based on the protocol for the SBC model. It is an everlasting equilibrium, and does not rely on any computational assumptions.

Due to space constrains, some proofs are omitted. *Full proofs and formal definitions can be found in the full version of this paper.*

2. SOLUTION CONCEPT

2.1 On Iterated Admissibility

As pointed out by Halpern and Teague [5], when considering information exchange tasks, requiring protocols to induce a Nash equilibrium is not enough to ensure stability (e.g., Shamir's scheme is a Nash equilibrium for $t > m$, but is unstable). Therefore, they were interested in protocols that are not only Nash equilibria, but are also iterated admissible. Recall that a strategy σ is said to be weakly dominated if there is another strategy τ that is always at least as good as σ, but is sometimes strictly better. Iterated admissible strategies are the ones surviving the iterative deletion of dominated strategies. In this section we show that iterated admissibility should not be used to distinguish "good" information exchange protocols from "bad" ones, and suggest the stronger notion of strict Nash equilibrium.

We show that many bad strategies are not ruled out by the iterated admissibility criterion. For example, we show that the protocol `talk-once`, in which every player broadcasts his share during the first round and then keeps silent forever, is actually iterated admissible. The finite version of this strategy was suggested by Halpern and Teague as an example of a bad solution. However, since they show that there are no rational secret sharing protocols with bounded number of moves, the finite version is problematic anyway, and it suffices to deal with its infinite version.

We define a model called `one-time-shares`, intended to match the one used in [5], though there are many details they do not make explicit. In this model, it is assumed that each player prefers learning the secret to not learning it, and secondarily, prefers that as few as possible of the other players learn it. A protocol proceeds in an infinite sequence of iterations, at the beginning of each, the dealer privately distributes fresh m-out-of-n Shamir shares of the secret to each of the players. During an iteration, the dealer does not take part in the protocol. Instead, the players run a randomized protocol amongst themselves by simultaneously broadcasting messages (Halpern and Teague additionally allow private communication between the players; we omit the private channels for simplicity). It is assumed that in every round the players either broadcast their share, or otherwise keep silent. Player i "learns" the secret in a specific protocol run if there is a round in which at least $m-1$ players other than i have broadcasted their share.

The following theorem shows that a large set of deterministic strategies denoted A_i, are iterated admissible. The set A_i contains pure strategies for player i that do not depend on the dealer's random tape. In other words, player i chooses his action for the next round by only considering which players have broadcasted in each of the previous rounds. The values of the shares dealt to player i and to the others are not taken into account.

THEOREM 1. *In the* `one-time-shares` *model for rational m-out-of-n secret sharing ($2 < m < n$), for every player i the strategies in A_i are iterated admissible. In particular,* $\texttt{talk-once}_i$ *is iterated admissible.*

Remark 1. Halpern and Teague do not specify how and when the game ends. In the suggested `one-time-shares` model, every run of every protocol is infinite. However, the theorem holds for different ending rules as well. For example, if:

- All players are required to send a quit message in order to end the game.
- The game ends when at least $m-1$ players have broadcasted their shares in the same round.

The theorem is proved by showing that for each candidate strategy τ "trying" to dominate a strategy $\sigma \in A_i$, there is a

"savior", a joint strategy of the others for which playing σ is preferable to playing τ. Therefore, σ is not dominated. For example, the strategy saving $\texttt{talk-once}_i$ from the silence strategy is the joint strategy of the other players in which each keeps silent during the first round, then reveals his share during the second round iff player i talked during the first round. More generally, the savior strategy waits to see if player i follows his prescribed strategy, then rewards or punishes him accordingly.

Since all the strategies in A_i survive the first order of deletions, and due to the fact that all the savior strategies used are also in A_i, we conclude that the strategies in A_i will never be deleted.

PROOF. Assume that every iteration consists of a single round. Let $h = (\mathbf{b}_1, ..., \mathbf{b}_t)$ be a list of boolean vectors of size n, $\mathbf{b}_s = (b_s^1, ..., b_s^n)$ where $b_s^i \in \{\texttt{TRUE}, \texttt{FALSE}\}$. We say that the boolean history of the game until round t agrees with h if for every round $s \leq t$ and player $j \in N$, j has broadcasted his share in round s iff $b_s^j = \texttt{TRUE}$.

Define two pure strategies, $\sigma_i^{h,+}$ and $\sigma_i^{h,-}$, based on h:

$\underline{\sigma_i^{h,+}}$:

In round s

- *for* $s \leq t$: if $b_s^i = \texttt{TRUE}$ broadcast your current share. Otherwise, keep silent.
- *for* $s = t + 1$: if the history of the game until round t agrees with h, broadcast your current share. Otherwise, keep silent.
- *for* $s \geq t + 2$: keep silent.

$\sigma_i^{h,-}$ is defined similarly, but in round $s = t + 1$ the player broadcasts his share only if the history *does not* agree with h, and keeps silent otherwise.

Let $\sigma_i \in A_i$, and assume for contradiction that there exists a strategy τ_i for player i that weakly dominates σ_i. In particular, there is a joint strategy of the other players $\boldsymbol{\tau}_{-i}$ for which $u_i(\tau_i, \boldsymbol{\tau}_{-i}) > u_i(\sigma_i, \boldsymbol{\tau}_{-i})$. Thus, there are random tapes for the dealer and players $\mathbf{r} = (r_D, r_1, ..., r_n)$, such that $u_i(R^{\tau_i}) > u_i(R^{\sigma_i})$, where R^{τ_i} is the run for which each player j follows τ_j and the random tapes are $\mathbf{r}$, and R^{σ_i} is a similar run for which player i follows σ_i instead of τ_i. Denote by t the first round for which the actions of player i in R^{τ_i} and R^{σ_i} differ.

Let $h = (\mathbf{b}_1, ..., \mathbf{b}_t)$ be a list of boolean vectors, where $\mathbf{b}_s$ is:

- *for* $s < t$: for $j \in N$ set $b_s^j = \texttt{TRUE}$ iff player j broadcasts his share in round s of R^{σ_i}.
- *for* $s = t$: for $j \neq i$ set $b_s^j = \texttt{FALSE}$; set $b_s^i = \texttt{TRUE}$ iff player i broadcasts his share in round t of R^{σ_i}.

One of the following must hold for R^{σ_i}:

1. There is a round $s < t$ in which at least m players broadcast their shares.
2. There is a round $s < t$ in which exactly $m - 1$ players other than i broadcast their shares, and Option (1) does not hold.
3. For all rounds $s < t$, at most $m - 2$ players other than i broadcast their shares.

We next show that in every possible case there is a "savior" strategy for σ_i.

If (1) *holds:* all the players learn the secret. Since similar messages are broadcasted in the first $t-1$ rounds of R^{τ_i} and R^{σ_i}, it holds that $u_i(R^{\tau_i}) = u_i(R^{\sigma_i})$, and a contradiction is reached.

If (2) *holds:* assume that the other players follow $\boldsymbol{\sigma}_{-i}^{h,-}$. Player i gets maximal payoff when the history until round t agrees with h: For such a history i learns the secret, but some of the others players do not, whereas if the history does not agree with h all players learn.

If player i follows τ_i, then with positive probability the history of the first t rounds does not agree with h (e.g., when the random tape used by player i agrees with r_i in all the positions used by τ_i in the first t rounds of R^{τ_i}). However, since $\sigma_i \in A_i$, when following σ_i the history always agrees with h. Thus, $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}^{h,-}) > u_i(\tau_i, \boldsymbol{\sigma}_{-i}^{h,-})$, and $\boldsymbol{\sigma}_{-i}^{h,-}$ "saves" σ_i.

If (3) *holds:* assume that the other players follow $\boldsymbol{\sigma}_{-i}^{h,+}$. Player i again gets maximal payoff when the history until round t agrees with h, since this is the only case allowing him to learn the secret. Hence, $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}^{h,+}) > u_i(\tau_i, \boldsymbol{\sigma}_{-i}^{h,+})$, and $\boldsymbol{\sigma}_{-i}^{h,+}$ "saves" σ_i.

None of the strategies in A_i are weakly dominated, because $\boldsymbol{\sigma}_{-i}^{h,+}$ and $\boldsymbol{\sigma}_{-i}^{h,-}$ save them. Since $\sigma_i^{h,+}, \sigma_i^{h,-} \in A_i$, they also survive the first iteration. We conclude that all the strategies in A_i survive the iterated elimination. □

An alternative concept: strict Nash equilibrium.

An informal explanation as to why `talk-once` is "bad" is that when the other players are following `talk-once`, player i gets the same payoff for staying silent, as he would have gotten had he also been following `talk-once`. In this case, a small change introduced to player i's belief is enough to make the silence strategy preferable: E.g., if player i thinks that the others would keep their silence with some arbitrarily small probability, he would prefer to follow the silence strategy himself. The fact that player i's strategy radically changes when making even minor modifications, suggests that the offered solutions is too fragile.

To rule out such bad solutions, we suggest the concept of strict Nash equilibrium, in which every player's strategy is a *strict* (and only) best response to the strategies of the others. We claim that strict equilibrium protocols do not suffer from the above problem: Since player i's strategy is a strict best response, every other strategy yields a payoff lower by at least c for some positive value c. When sufficiently small changes (as a function of c) are introduced to the rules of the game (e.g., slight changes of the utilities or of the set of possible actions) or to i's belief, i's best response is still close to his original strategy.

The notion of strict equilibrium is stronger than Nash equilibrium, and since it ensures the uniqueness of a player's best response, it also implies iterated admissibility.

2.2 On the Backward Induction Process

Previously suggested rational secret sharing schemes make use of cryptographic primitives (e.g., see [1, 10]). However, unlike standard cryptographic protocols, these schemes may run for an exponential number of rounds (at least with an exponentially small probability). The key problem is that there is a bound b (possibly very large), such that after b rounds any player can break the cryptographic primitives

used in the first round and reveal the other players' shares encoded by them. Therefore, round b is essentially the last, and players have no incentive to cooperate if it is reached since they no longer fear future punishment. Consequently, round $b-1$ is now essentially the last round, and players deviate for the same reason. The process continues in this way backwards in time, thus it is called backward induction, showing that players are better off keeping silent in rounds $b-2, b-3, ..., 1$ as well.

The backward induction process in computational settings, where presumably we are not concerned with the protocol's stability in rare events, is as problematic as in the standard Game Theoretic settings, since it *causes exponential events to be amplified* (e.g., the instability of the cryptographic protocols when they reach their b^{th} round causes them to be unstable from round 1).

Everlasting equilibrium.

In this paper we take a different approach and offer non-cryptographic rational secret sharing schemes. The schemes are immune to the backward induction process since they satisfy the additional property that *after any history* on the equilibrium path (i.e., a history that can be reached by the protocol*),* following the protocol is still a Nash (strict Nash, ε-Nash) equilibrium. We call such protocols everlasting (strict everlasting, ε-everlasting) equilibria.

The aforementioned property holds trivially for any (*exact*) Nash equilibria: If a player can get a higher payoff by deviating - his strategy is not a best response. However, it does not necessarily hold for approximated equilibria, such as ε-Nash equilibria, since the ignored "ε" term may indicate that in some rare situations the prescribed strategies are far from optimal. For example, the cryptographic protocols mentioned above are close to equilibrium, but after any history of length $b-1$, the strategies they prescribe are far from being best responses.

Remark 2. The concept of everlasting equilibrium resembles the Game Theoretic notions of subgame perfect or sequential equilibria, but is weaker. In a subgame perfect equilibrium, the prescribed strategies must be best responses after any history, even after histories that cannot be reached by the protocol. Such protocols ensure that there are no "non-credible threats", in the sense that carrying them out will harm the player making the threat (e.g., a beggar threats to commit suicide if you do not give him charity). An everlasting equilibria only requires the prescribed strategies to be best responses after any history on the equilibrium path, and it eliminates "non-credible promises" (e.g., players' non-credible promises to cooperate in the last round).

3. OUR SETTINGS

We review the models for rational MPC and rational secret sharing used in the paper.

3.1 Settings for Rational MPC

In rational MPC a set of players $N = \{1, ..., n\}$ each holding an input are interested in evaluating an n-ary function $f : \mathbf{X} \to Y$ ($\mathbf{X} \subseteq \times_{i \in N} X_i$ for some sets X_i) with a finite range. Our input as protocol designers is the function f, the distribution over inputs $\mathcal{D}$, and players' preferences given as utility functions $(u_i)_{i \in N}$. Recall that utility functions associate numeric values to outcomes of the game (in our case, an outcome consists of the players' inputs, and the sequence of actions taken by them), the value $u_i(o)$ is player i's payoff if outcome o was reached. Actually, as discussed later, we only require partial information about the utility functions and the distribution. We should then output a game and "rational" strategies (randomized algorithms) for the players allowing all of them to learn $f(\mathbf{x})$. We stress that the players' utility functions are predetermined and cannot be changed.

We suggest a computing game for f, with respect to $(u_i)_{i \in N}$ and $\mathcal{D}$, that proceeds in a sequence of rounds. In every round, players are allowed to broadcast *any* finite binary string of their choice. A player can leave the game in any round by broadcasting a quit sequence and outputting his guess of $f(\mathbf{x})$. Players observe the actions taken by the others in previous rounds, but do not view their guesses of the secret.

If an SBC is assumed, the broadcasts in every round are *simultaneous*, and the game is called a *simultaneous* computing game (SCG) and is denoted $\Gamma_f^{\mathcal{D},(u_i)_{i \in N}}$ (Γ_f for short). Otherwise, an NSBC is assumed, and only a single player may broadcast in every round. Such a game is called a *non-simultaneous* computing game (NSCG) and is denote $\bar{\Gamma}_f^{\mathcal{D},(u_i)_{i \in N}}$ ($\bar{\Gamma}_f$ for short). In an NSCG, we make no assumptions regarding the NSBC's behavior when two or more players try to broadcast at the same time. In such cases, some or all players may get partial information about the messages. The rest of the definitions are formulated for SCGs, but can be similarly formulated for NSBCs.

A protocol $\boldsymbol{\sigma}$ for Γ_f is an assignment of randomize strategies to players. We say that $\boldsymbol{\sigma}$ computes f if it almost always ends for every set of inputs $\mathbf{x} \in \mathbf{X}$ used by the players, and whenever it ends all players output $f(\mathbf{x})$.

3.2 Settings for Rational Secret Sharing

A rational (strict rational, ε-rational) secret sharing scheme consists of a dealer's algorithm for issuing shares, and a protocol allowing the players to reconstruct the secret. We make the following two requirements: First, as in the classical settings, the shares should be such that any m or more determine the secret, but less than m convey no information about the secret. Second, we require the reconstruction protocol to be an *everlasting* (*strict everlasting*, *ε-everlasting*) *equilibrium.* If an SBC is used we call the scheme a simultaneous rational scheme, otherwise it is a *non*-simultaneous rational scheme.

Rational MPC and rational secret sharing are closely related. In a rational secret sharing scheme the dealer equips players with inputs to some non-trivial function f taken from a known distribution. Then, the players interested in reconstructing the secret run a rational protocol for computing f in the game Γ_f. Therefore, *every rational secret sharing scheme requires a rational MPC of some function.*

3.3 Assumptions on the Utility Functions

In the next sections we assume that each utility function u_i satisfies some or all of the below properties. We say that a player retrieves the designated value (the secret or the function's value) when outcome o is reached, if according to o the player quits and outputs the correct value. Let o and o' be two possible outcomes of the game, and let retrieve(o) be the set of players retrieving the value when o is reached:

1. $u_i(o) > u_i(o')$ whenever $i \in$ retrieve(o) and $i \notin$ retrieve(o') (players prefer to learn).

2. If $i \in \mathsf{retrieve}(o)$ then $u_i(o) > u_i(o')$ whenever $\mathsf{retrieve}(o') = N$ and $\mathsf{retrieve}(o) \neq N$ (players prefer to learn while others do not).
3. If $i \in \mathsf{retrieve}(o)$ then $u_i(o) = g(|\mathsf{retrieve}(o)|)$ for some $g : \{0, 1, ..., n\} \to \mathbb{R}$ (the payoff is determined by the *number* of players learning).

If the first property is satisfied, we say that the utility functions are learning preferring. If all three hold, the utilities are strictly competitive. Our negative results assume strictly competitive utilities, whereas the positive results only use the learning preferring property.

3.4 The Linger Avoiding Assumption

In the following sections we assume that players will only follow equilibria protocols that compute f and prescribe linger avoiding strategies, i.e., strategies in which players quit immediately after learning the value. Since by the definition of an equilibrium, no player can gain from deviating, and in particular no player can prevent others from learning the value, *we are only requiring players to quit when they cannot gain from staying in the game.* In particular, in this case no player can confuse the others by staying in the game and broadcasting fictitious messages. Clearly, players are never worse off quitting in such situations, and may even be better off at times (e.g., if for some reason the game ends after this round).

This technical assumption is needed when in search of protocols satisfying the strictness property: When a player runs a non-linger avoiding strategy, the "linger avoiding version" of his strategy is another best response. Thus, no strict equilibria can be found. To overcome this problem, we use the weaker notion of a equilibrium with respect to linger avoiding strategies and only require each player's strategy to be: (*i*) a best response, (*ii*) strictly better than any *linger avoiding* strategy that acts differently on the equilibrium path.

4. IMPOSSIBILITY RESULTS

We show that for *every* non-constant function f with *finite domain* there is no linger avoiding Nash equilibrium protocol that computes f without a revelation point, as discussed in the Introduction and defined below. Informally speaking, this implies that there is no "reasonable" Nash equilibrium protocol for rational MPC.

4.1 Transcripts Tree and Revelation Points

We formalize the concept of revelation points used to rule out "bad" solutions via the notion of a transcripts tree. A run R of $\boldsymbol{\sigma}$ is a pair $R = (\mathbf{x}, \mathbf{r})$, where x_i is the private input of player i and r_i is his random tape. A transcript of $\boldsymbol{\sigma}$ is a sequence $\mathbf{m} = (\mathbf{m}_1, ..., \mathbf{m}_t)$ of messages broadcasted by the players during the first t rounds of a possible run R of $\boldsymbol{\sigma}$. That is, for every $s \leq t$, $\mathbf{m}_s = (m_s^1, ..., m_s^n)$ and m_s^i is a finite binary string broadcasted by player i in round t of R. In such a case we say that R explains $\mathbf{m}$, and write $\mathsf{m}(R, t) = \mathbf{m}$.

Note that since $\boldsymbol{\sigma}$ is a randomized algorithm, it may have various transcripts of the same length. Denote by $M(\boldsymbol{\sigma})$ the set of all transcripts of $\boldsymbol{\sigma}$. We view the elements of $M(\boldsymbol{\sigma})$ as vertices of a tree called the transcripts tree: The tree's root is the empty history, and the transcript $\mathbf{m}$ of length t is the parent of the transcript $\mathbf{m}'$ of length $t+1$ if $\mathbf{m}$ is a prefix of $\mathbf{m}'$.

We say that player i learns (knows) $f(\mathbf{x})$ after round t of run R (or, equivalently, after transcript $\mathbf{m} = \mathsf{m}(R, t)$) given input x_i, if given player i's view there is only one possible value for $f(\mathbf{x})$. In other words, there exists $y \in Y$ such that $f(\mathbf{x}') = y$ for every $\mathbf{x}'$ with $x_i' = x_i$ for which the protocol $\boldsymbol{\sigma}$ ran with the input $\mathbf{x}'$ can yield the transcript $\mathbf{m}$. Finally, a revelation point of $\boldsymbol{\sigma}$ is defined to be a transcript $\mathbf{m}$ that satisfies both of the following requirements: First, there exists a player i and an input x_i such that i does not know $f(\mathbf{x})$ after $\mathbf{m}$ given x_i. Second, *any* player i knows $f(\mathbf{x})$ after *any* child of $\mathbf{m}$ given *any* possible input x_i.

4.2 Impossibility Results for Rational MPC

We are now ready to state the main result of this section. Note that the impossibility results are formulated for the SBC model, but hold for the NSBC model as well, since an NSBC can be viewed as a special SBC: in this kind of SBC only one player sends a "real" message in each round, the others are sending a special "not broadcasting" message that can be ignored.

THEOREM 2. *Let f be a non-constant function with a finite domain and any number of players, and let Γ_f be an SCG for f with respect to strictly competitive utility functions. There is no linger avoiding protocol for Γ_f that computes f and does not have a revelation point.*

We start by sketching the proof of the above theorem, the formal proof follows. We first note that a revelation point of a linger avoiding protocol is a vertex in the transcripts tree that has children but not grandchildren. The proof constructs a path in the tree leading to such a vertex, and uses the following claim: For every vertex $\mathbf{p}$ in the transcript tree, and every possible input vector $\mathbf{x}$, either players learn after all children of $\mathbf{p}$ when given $\mathbf{x}$, or they do not learn after any child of $\mathbf{p}$ when given $\mathbf{x}$.

The claim clearly holds for the 2 players case: Suppose that players learn the value after the child $\mathbf{m}$ of $\mathbf{p}$, but do not learn it after the child $\mathbf{m}'$. Since it is possible that after reaching $\mathbf{p}$ player 1 will choose to act according to $\mathbf{m}'$ while player 2 decides to act according to $\mathbf{m}$, player 1 may learn the secret alone. Since we assumed that the protocol allows all players to learn the value, we have reached a contradiction. To show that a similar claim holds for any number of players we use a hybrid argument.

We now turn to find the first vertex on the branch leading to the revelation point. Before the game begins, players do not know which input vector was selected. Assume that the game ends for some possible inputs vector $\mathbf{x}$ after $\mathbf{m}'$ was reached, and denote his parent by $\mathbf{p}$. If $\mathbf{p}$ has no grandchildren, then $\mathbf{p}$ itself is a revelation point. Otherwise, let $\mathbf{m}$ be child of $\mathbf{p}$, giving it grandchildren. Using the above claim, since all players learn $f(\mathbf{x})$ after $\mathbf{m}'$ given inputs $\mathbf{x}$, they all learn it after $\mathbf{m}$ as well. Thus, if the protocol proceeds past $\mathbf{m}$, players know that they were not given the inputs $\mathbf{x}$. $\mathbf{m}$ is now our first landmark in the way to the revelation point.

We proceed by *induction*: The vertex $\mathbf{m}$ is viewed as a beginning of a new game, and the same process is applied. Due to the *finiteness* of the inputs set, and the fact that we "lose" at least one input in each such iteration, it can be concluded that the process can only be used a finite number of times. Since the process only ends when a revelation point is reached, the claim holds.

PROOF. Let $\boldsymbol{\sigma}$ be a linger avoiding protocol. For $\mathbf{m} \in M(\boldsymbol{\sigma})$ denote by C_m the set of pairs $(i, \mathbf{x})$ such that $i \in N$,$\mathbf{x} \in \mathbf{X}$ and i does not know $f(\mathbf{x})$ given x_i.

Let $\mathbf{m}_0$ be the empty transcript. $C_{\mathbf{m}_0} \neq \phi$, otherwise every player can always deduce $f(\mathbf{x})$ by himself, and thus f is constant. Choose $\mathbf{x}_1 \in \mathbf{X}$ and $j_1 \in N$ for which $(j_1, \mathbf{x}_1) \in C_{\mathbf{m}_0}$. Since the protocol almost always ends, there is a run $R' = (\mathbf{x}_1, \mathbf{r}')$ of $\boldsymbol{\sigma}$ for which $\mathbf{m}'_1 = \mathsf{m}(R', t)$ is a descendant of $\mathbf{m}_0$ for some t, and all players know the designated values after round t of R'. Assume that t was chosen to be minimal, that is, some players do not know the value after round $t-1$ of R'.

Denote $\mathbf{m}'_1$'s parent by $\mathbf{p}$. If every child $\mathbf{m}$ of $\mathbf{p}$ satisfies $C_{\mathbf{m}} = \phi$, then $\mathbf{p}$ is a revelation point. Otherwise, we show that there is a child $\mathbf{m}_1$ of $\mathbf{p}$ such that $C_{\mathbf{m}_1} \neq \phi$ and $C_{\mathbf{m}_1} \subsetneqq C_{\mathbf{m}_0}$: Start from any child $\mathbf{m}''_1$ of $\mathbf{p}$ for which $C_{\mathbf{m}''_1} \neq \phi$. If $(j_1, \mathbf{x}_1) \notin C_{\mathbf{m}''_1}$, the transcript $\mathbf{m}_1 = \mathbf{m}''_1$ satisfies our requirement. Otherwise, there is a run $R'' = (\mathbf{x}_1, \mathbf{r}'')$ of $\boldsymbol{\sigma}$ explaining $\mathbf{m}''_1$ for which j_1 does not know the value after round t.

Denote $\mathbf{r}' = (r'_1, ..., r'_n)$, $\mathbf{r}'' = (r''_1, ..., r''_n)$, and for $i \in [n+1]$ let $\mathbf{r}^i$ be the hybrid $(r'_1, ..., r'_{i-1}, r''_i, ..., r''_n)$. Due to the fact that both R' and R'' explain $\mathbf{p}$, so does $R^i = (\mathbf{x}_1, \mathbf{r}^i)$. This is shown by induction on the length of $\mathbf{p}$, since each party's public messages only depend on his own random tape and the previous messages sent.

Since $\mathbf{r}^1 = \mathbf{r}''$ and $\mathbf{r}^{n+1} = \mathbf{r}'$, there is $i \in N$ such that after round t of run R^i some players still do not know the value, but after round t of R^{i+1} all players know it. Player i being the only one assigned different random tapes by R^i and R^{i+1}, is the only player taking a (possibly) different action in round t of R^i and R^{i+1}. Since the other players make the exact same moves, we conclude that i knows the value after round t of R^i, just as he knows it after round t of R^{i+1}.

We next show that player i must have learned the value during round t, and not before: Since $\boldsymbol{\sigma}$ is linger avoiding, player i quits immediately after learning the value. Had player i learned the value during a previous round, the massage broadcasted by him in round t is independent of his random tape: A quit message is broadcasted if i learned during round $t-1$, and an empty messages is broadcasted if he learned before round $t-1$. Consequently, all players learn the value after round t of R^i, just as they learn it after round t of R^{i+1}. Since this contradicts our assumption about R^i, we deduce that player i indeed learns during round t of R^i. By choosing $\mathbf{m}_1 = \mathsf{m}(R^i, t)$, we get $C_{\mathbf{m}_1} \neq \phi$ and $(i, \mathbf{x}_1) \in C_{\mathbf{m}_0} \backslash C_{\mathbf{m}_1}$.

A sequence of transcripts, $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, ...$, such that $N \times X \supseteq C_{\mathbf{m}_0} \supsetneqq C_{\mathbf{m}_1} \supsetneqq C_{\mathbf{m}_2} \supsetneqq ...$ is built using the same arguments. Since the set $N \times X$ is finite, the sequence ends and a revelation point is found. □

Remark 3. Theorem 2 does not imply that there is an efficient algorithm for finding revelation points. However, since a revelation point $\mathbf{m}_{rev}$ does exist, player i may prefer to deviate when $\mathbf{m}_{rev}$ is reached.

The next corollary shows that since there are no "reasonable" Nash equilibria protocols for rational MPC, there are no strict ones as well:

COROLLARY 1. *Let f be a non-constant function with a finite domain and any number of players, and let Γ_f be an SCG for f with respect to strictly competitive utility functions. There is no **strict** Nash equilibrium protocol with respect to linger avoiding strategies for Γ_f that computes f.*

For *two* players games, a stronger result can be obtained. The key observation is that in such games, both players must learn the secret together (during the same round). Otherwise, the player learning the secret first may pretend to be holding a fictitious input, and thus cause the other player to "learn" a wrong value with some positive probability.

COROLLARY 2. *Let f be a **two** players non-constant function with a finite domain, and let Γ_f be an SCG for f with respect to strictly competitive utility functions. There is no **Nash equilibrium** protocol for Γ_f that computes f.*

CLAIM 1. *Let f be a two players non-constant function with a **countable domain**, and let $\overline{\Gamma}_f$ be an **NSCG** for f with respect to strictly competitive utility functions. There is no Nash equilibrium protocol for $\overline{\Gamma}_f$ that computes f.*

4.3 Impossibility Results for Rational Secret Sharing

Recall that rational secret sharing requires rational MPC of some non-constant function. In light of Theorem 2, there is no "reasonable" (simultaneous or non-simultaneous) rational secret sharing scheme for any set of secrets Y with $|Y| > 1$, in which the dealer assigns shares taken from finite sets. In particular, there are no such m-out-of-n *strict* rational secret sharing schemes, and no such 2-out-of-n rational schemes.

5. A STRICT RATIONAL SCHEME WITH UNBOUNDED SHARES

Fortunately, the impossibility results of the Section 4 rely heavily on the finiteness of the function's domain. It turns out that by allowing the shares to be taken from unbounded domains, rational secret sharing schemes can be obtained. In this section we suggest such a scheme that uses an SBC and satisfies the strictness property. We first describe a scheme for 2-out-of-2 secret sharing, and then show how to extend it to m-out-of-n for any $2 \leq m \leq n$.

The 2-out-of-2 Case.

The basic idea is that the shares assigned to the player are lists of possible secrets (elements of Y), such that one of the lists is a strict prefix of the other. We call the player receiving the shorter share the "short player", and the player with the longer share the "long player". *Players are not informed whether their share is long or short.*

In order to construct the desired shares the dealer first selects the index of the definitive iteration ℓ (which also determines the size of the shorter list), and then the number of extra elements in the longer list, denoted d. Both ℓ and d are chosen according to a geometric distribution with parameter β, where β depends on the utility functions. We will discuss how β is chosen later. The dealer then chooses a random list of possible secrets of size $d + \ell - 1$, such that its ℓ^{th} element is the real secret. The complete list is given to one of the players, while the other player gets only a prefix of the list, containing all elements in positions prior to ℓ.

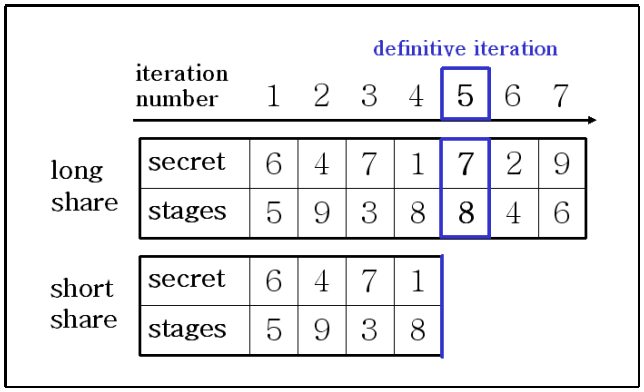

Figure 1: Possible shares assigned by the dealer
In this example the set of secrets is $S = \{1, ..., 10\}$, the real secret is $y = 7$, the definitive iteration is $\ell = 5$, and the number of extra elements is $d = 3$.

The shares are designed so that the first possible secret to appear only in the long list is the real secret. In order to reconstruct the secret, players are expected to broadcast the next secret in their list in every iteration, and keep silent after their list ends. The first possible secret broadcasted by only one player is assumed to be the real secret, and the iteration in which it is revealed (iteration ℓ) is called the definitive iteration. Note that a player's behavior does *not* depend on messages broadcasted by the others (aside from when he leaves the game), but is determined by his share.

This basic idea has several weak points. One obvious problem is the ability of the short player to detect the definitive iteration before it is carried out. Indeed, when the short player runs out of secrets to broadcast, he knows that the next iteration is the definitive one. In such a case the short player may broadcast a fictitious secret instead of keeping quiet. With a (small) positive probability he will be able to guess the next element in the long player's list, causing the long player to believe that the secret was not yet revealed.

To prevent the short player from deviating during the definitive iteration, we divide every iteration into a number of separate stages. The number of stages varies from iteration to iteration, and is again chosen according to the geometric distribution with parameter β. Each player then receives the number of stages in each iteration described in his list. We ask players to broadcast only during the last stage of each iteration. Now, the short player knows when the definitive iteration is reached, but does not know the exact number of stages in the iteration, whereas the long player knows the length of all iterations, but is unable to identify the definitive iteration before it is carried out. An example for the shares distributed by the dealer's algorithm (as described so far) is given in Figure 1.

Another weak point of the basic idea is the possibility that most or all future secrets in the list have the same value, allowing the players to guess the secret. This can be prevented by masking every element in the list using a different random mask. Shares of the random masks are dealt to the players. In iteration t, players are required to broadcast their share of the mask that will be used in iteration $t + 1$.

In order to prevent players from broadcasting false information (such as a fictitious mask share), we equip each with authentication information. Using the information, a player can verify the authenticity of the messages broadcasted by the others, and prove the authenticity of messages sent by him.[3]

The General Case.

To generalize the above to an m-out-of-n secret sharing scheme, the long list is given to all but one player. Since now a subset of m or more players that does not contain the short player is unable to identify the definitive iteration after it is carried out, we add a boolean indicator (via secret sharing) showing whether the current iteration is definitive.

Formal description of the dealer's and players' algorithms, as well as some additional notes, can be found in Figures 2 and 3.

Remark 4. The described protocol is susceptible to coalitions. For example, if the short player colludes with one of the long players, together they can learn the secret before the definitive iteration is carried out.

Protocol Analysis.

Theorem 3 (below) shows that the suggested scheme is a *strict* rational secret sharing scheme with respect to learning preferring utility functions under the two following conditions (note that we do not need to assume that players prefer to learn the secret without the others):

First, β should be chosen to be small enough. For example, in the case of strictly competitive utilities, the greater the ratio between the payoff for learning the secret alone and learning with the others, the smaller β must be in order to prevent players from guessing the definitive iteration and deviating. As β is getting smaller, the probability of deviating in the wrong iteration, thus causing the game to end, increases.

Second, players must have an initial incentive to cooperate: We cannot expect a player to participate in a sharing scheme if he can a-priori guess the secret with a sufficiently high probability. If $b \in Y$ is the element with highest probability according to $\mathcal{D}$, then every player can guess the secret with probability at least $\mathcal{D}(b)$. Therefore, we must assume that $\mathcal{D}(b)$ is sufficiently small.

The theorem below holds for $\beta < \beta_0$ and $\mathcal{D}(b) < c_0$. The values of β_0 and c_0 are functions of the utilities, their values are calculated in the full version of this paper. The theorem's proof is based on the observation that a player cannot learn anything (*information theoretically*) from non-definitive iterations, since the information broadcasted in such iterations was randomly chosen. Therefore, after any history, players are still better off following the protocol, and there is no essential bound on the length of the protocol.

THEOREM 3. *Let Y be a finite set of secrets with distribution $\mathcal{D}$, and let $(u_i)_{i \in N}$ be learning preferring utility functions. If $\mathcal{D}(b) < c_0$, then for $\beta < \beta_0$ and for all $2 \le m \le n$, the scheme described above is a simultaneous **strict** rational m-out-of-n secret sharing scheme for Y with respect to linger*

[3]For example, this can be done using the following method (see [15, 13]): If player i's true information is $x \in \mathbb{F}$, then $s_i, b_i \in \mathbb{F}$, $b_i \neq 0$, are chosen at random and we set $c_i = b_i \cdot x + s_i \in \mathbb{F}$. The value s_i (the *tag*) is given to i. The other players each get b_i and c_i (the *hash function*). Player i is required to broadcast s_i in order to prove that x is his true information. The other players can then verify with high probability by checking that $c_i = b_i \cdot x + s_i$.

Dealer(y, β)

Let $\mathbb{F} = GF(p)$ for $p \geq |Y|$ prime, and identify each element of the secrets set Y with an element of $\mathbb{F}$. Denote by $\mathcal{G}(\beta)$ the geometric distribution with parameter β.

Create the list of possible secrets:

- Select $\ell, d \sim \mathcal{G}(\beta)$. Iteration ℓ is the definitive one and $L = \ell + d - 1$ is the size of the full list of secrets.
- Select at random a list of size L of possible secrets (elements of Y), such that its ℓ^{th} element is y.

Create shares: Create n vectors, one of length $\ell - 1$ and the others of length L. Each vector cell corresponds to an iteration of the reconstruction protocol and consists of the following elements:

- **Stages**: The number of stages in the iteration chosen according to $\mathcal{G}(\beta)$.
- **Mask**: An m-out-of-n Shamir share of a randomly chosen element of $\mathbb{F}$ used to mask the *next* secret.
- **Masked secret**: An element of $\mathbb{F}$ obtained by summing, over $\mathbb{F}$, the corresponding element in the secrets list and the mask shared between the players in the *previous cells*.
- **Indicator**: An m-out-of-n Shamir share of a boolean indicating whether this iteration is the definitive one.
- **Authentication information**: A "tag" allowing the player to prove the authenticity the previous elements in this cell, and "hash functions" allowing him to check the authenticity of elements in the corresponding cells of the other vectors with probability at least $1 - \beta$.

  An additional cell is added to the beginning each vector ("cell 0"). The cell contains an m-out-of-n Shamir share of a randomly chosen mask to be used during the first iteration, and authentication information for it.

Assign shares: Choose a random assignment of vectors to players.

Figure 2: The dealer's algorithm

avoiding strategies. It has expected running time $O(\frac{1}{\beta^2})$*, and expected share size* $O(\frac{1}{\beta} \log \frac{1}{\beta})$.

Remark 5. The expected running time of the suggested protocol depends on the utility functions. For example, in the case of strictly competitive utility functions assigning payoff 0 to players that do not learn, the expected running time is a function of the ratio between the payoff for learning alone and the payoff for learning with the others. This property is inherent: suppose that there is an algorithm with expected running time independent of the ratio. For a large enough ratio, a player is better off guessing the last round of the protocol and deviating.

Player$_\mathbf{i}$ $(share)$

Set `secret` $\leftarrow$ `FALSE` and `cheat` $\leftarrow$ `FALSE`.

Repeat until `secret` = `TRUE` **or** `cheat` = `TRUE`

If your share ended: Keep silent. If someone has broadcasted, `secret` $\leftarrow$ `TRUE`.

If your share did not end: use the corresponding cell of *share* to check whether this is the last stage of this iteration.

- If this is ***not*** the last stage: Keep silent. If someone broadcasted `cheat` $\leftarrow$ `TRUE`.
- If this ***is*** the last stage:
  - Broadcast the masked secret, tag, and shares of the random mask and indicator, as they appear in the corresponding cell of *share*.
  - If more than a single player did not broadcast, or if some messages do not pass the authenticity check (the tags and hash functions do not match), `cheat` $\leftarrow$ `TRUE`.
  - If all but a single player broadcasted, or if the reconstructed indicator shows that the iteration is definitive, `secret` $\leftarrow$ `TRUE`.

Leave the game: Quit and output the current possible secret (obtained by subtracting the mask reconstructed using the shares broadcasted in the *previous* iteration from the last masked secret broadcasted).

Figure 3: Player i's reconstruction protocol

6. AN $\mathcal{E}$-RATIONAL SCHEME FOR THE NSBC MODEL

In this section we describe an ε-rational m-out-of-n secret sharing scheme for the NSBC model, based on the SBC scheme suggested in Section 5. The straightforward adaptation of the previous scheme is having the players broadcast one after the other in a *predefined order*, instead of simultaneously. In other words, every simultaneous stage is replaced by n non-simultaneous rounds, each allows one of the players to broadcast.

However, the resulting scheme has a flaw: If the short player happens to be the first to broadcast according to the predefined order, then the first stage of the definitive iteration starts with a silent round. The long players can use the silent round as an indication that the definitive iteration was reached, and quit while outputting the next unmasked secret. In such a case the short player stays ignorant.

In order to overcome the problem, we select a *different broadcasts order* for every iteration. The broadcasts orders are determined by permutations selected independently at random by the dealer. Each player receives the permutations for every iteration in his list. The player chosen to be the last to broadcast in the definitive iteration is given the short share.

Claim 1 implies that there are no (exact) rational secret sharing schemes for the NSBC model (at least for 2 players), even when shares are taken from an unbounded domain. In-

deed, the suggested scheme is not an exact rational scheme, since the short player might broadcast a fictitious secret instead of keeping quiet during the definitive iteration. However, the scheme is ε-rational when we use an authentication mechanism ensuring that attempts to authenticate fictitious messages will fail with probability at least $1 - \frac{\varepsilon}{U_{max}}$, where U_{max} is an upper bound on the payoffs that the players may receive. Note that ε can be made arbitrarily small at the price of having longer shares (more authentication data). Specifically, ε is exponentially small in the share sizes.

THEOREM 4. *Let Y be a finite set of secrets with distribution $\mathcal{D}$, and let $(u_i)_{i\in N}$ be learning preferring utility functions. If $\mathcal{D}(b) < c_0$, then there exists $\beta_0' > 0$ (a function of the utility function, the size of the secrets set and the number of players) such that for $\beta < \beta_0'$ and for all $2 \leq m \leq n$, the scheme described above is a non-simultaneous **ε-rational** m-out-of-n secret sharing scheme for Y. It has expected running time $O(\frac{n}{\beta})$, and expected share size*
$O\left(\frac{n \lg n}{\beta}(\log \frac{1}{\beta} + \log \frac{U_{max}}{\varepsilon})\right)$.

Remark 6. The described protocol is susceptible to existence of a malicious player: Such a player can cause the others to output a wrong value by simply aborting prematurely. However, the deviating player will not be able to learn the secret himself. Since we assume that all players are rational individuals that prefer to learn above all else, there will never be an incentive to such behavior.

Remark 7. By simply truncating the shares to size T and adding a cell containing the real secret to the end of each vector, we get a scheme with bounded shares length. In particular, if T is chosen such that the game ends after the first T iterations with probability at least $1 - \varepsilon$ (i.e., $(1 - \beta)^T < \varepsilon$), then the suggested scheme is 2ε-rational. Note, however, that the resulting scheme is not a 2ε-everlasting equilibrium, and is susceptible to backward induction.

7. DISCUSSION AND OPEN PROBLEMS

This paper raises several new open problems. The first is that of further exploring the Game Theoretic considerations one needs to take into account when designing information exchange protocols. Other, more concrete problems, are finding ε-everlasting equilibria schemes with shares taken from bounded domains (we have only shown that no such *exact* equilibria are possible), obtaining good everlasting schemes that are also coalition-proof, and characterizing what MPC problems have such protocols. Note that we offer such cryptographic results, under computational assumptions, in the subsequent work [7].

Acknowledgments

We thank Or Meir for many helpful comments.

8. REFERENCES

[1] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. *In Proceedings of the 25th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 53-62, 2006.

[2] D. Boneh and M. Naor. Timed commitments, *Advances in Cryptology - CRYPTO 2000*, *Springer LNCS 1880*, pages 236-254, 2000.

[3] J. Garay and M. Jakobsson. Timed Release of Standard Digital Signatures, *In Proceedings of Financial Cryptography*, LNCS 2357, pages 168-182, Springer, 2002.

[4] S. D. Gordon and J. Katz. Rational Secret Sharing, Revisited. *Security and Cryptography for Networks (SCN)*, pages 229-241, 2006.

[5] J. Halpern and V. Teague. Rational Secret Sharing and Multiparty Computation. *In Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 623-632, 2004.

[6] S. Izmalkov, S. Micali, and M. Lepinski. Rational Secure Computation and Ideal Melearnchanism Design. *In Proceedings of the 46th IEEE Symposium of Foundations of Computer Science (FOCS)*, pages 585-595, 2005.

[7] G. Kol and M. Naor. Cryptography and Game Theory: Designing Protocols for Exchanging Information. *In the Proceedings of the 5th Theory of Cryptography Conference (TCC)*, pages 317-336, 2008.

[8] M. Lepinski, S. Micali, C. Peikert, and A. Shelat. Completely Fair SFE and Coalition-Safe Cheap Talk. *In Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 1-10, 2004.

[9] M. Lepinski, S. Micali, and A. Shelat. Collusion-Free Protocols. *In Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 543-552, 2005.

[10] A. Lysyanskaya and N. Triandopoulos. Rationality and Adversarial Behavior in Multi-Party Computation. *Advances in Cryptology - CRYPTO 2006*, pages 180-197, 2006.

[11] M. Osborne and A. Rubinstein. A Course in Game Theory, *MIT Press*, 1994.

[12] B. Pinkas. Fair Secure Two-Party Computation. *Advances in Cryptology - Eurocrypt 2003*, pages 87-105, 2003.

[13] T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multiparty Protocols with Honest Majority. *In Proceedings of the 21th Annual ACM Symposium on Theory of Computing (STOC)*, pages 73-85, 1989.

[14] A. Shamir. How to share a secret. *Communications of the ACM*, volume 22, pages 612-613, 1979.

[15] M. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, volume 22, pages 265-279, 1981.