Games For Exchanging Information[*]

Gillat Kol[†]　　Moni Naor[§]

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.

[†] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100 Israel. Email: gillat.kol@weizmann.ac.il.

[§] Incumbent of the Judith Kleeman Professorial Chair, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100 Israel. Email: moni.naor@weizmann.ac.il.

[*] Research supported by a grant from the Israel Science Foundation.

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 [7]. 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 [7], 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. That is, 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 protocol and keep their information to themselves. In order demonstrate the problem, recall the m-out-of-n secret sharing scheme due to Shamir [17]: 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. For $m = n$ 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 $m < n$, 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

[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.

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 [7], Gordon and Katz [6], Abraham et al. [1], and Lysyanskaya and Triandopoulos [12]. 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 [12] 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 [6, 1, 12] 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. [10, 11] and Izmalkov et al. [8] 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

[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, 14] for recent work). Note, however, that such results do not take into account the rationality consideration that we use in this work.

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 [7] and adopted by [6, 1, 12], 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 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.

Our Settings (Section 3).

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.[3] 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 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 that computes f without a revelation point. 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. For a comparison of our impossibility results to the results offered by Halpern and Teague [7] see Remark D.1.

A Strict Rational Secret Sharing 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 Secret Sharing 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.

[3] Several stable protocols with an "on-line dealer" (a dealer that is involved in reconstruction protocol) were suggested in [7, 6, 1], all having points similar to our revelation point. However, those points could not be recognized by the players. In the protocols of [6, 1], the dealer chooses the revelation time and hides it from the players. In [7], the broadcast channel is not the only mean of communication and additional private channels are used, making the revelation point undetectable. Our claim is for protocols with no on-line dealer and no private channels.

In this section we offer such an 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, is an everlasting equilibrium, and does not rely on any computational assumptions.

For formal definitions of Game Theoretic concepts, please see Appendix B.

2 Solution Concept

2.1 On Iterated Admissibility

As pointed out by Halpern and Teague [7], 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 $m < n$, 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, and that 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.

On Iterated Admissibility. We first note that the notion of iterated admissibility was criticized within the Game Theory community, see discussion in Appendix A.1. Furthermore, Theorem A.3 shows that many bad strategies are not ruled out by the iterated admissibility criterion. For example, we show that the strategy `talk-once` (a player broadcasts his share during the first round and then keeps silent forever), whose *finite* version was given by Halpern and Teague as an example of a bad solution, is actually iterated admissible. Note that since they show that there are no rational secret sharing protocols with bounded number of moves, the bounded version of `talk-once` is problematic anyway, and it suffices to deal with its infinite version.

The theorem is proved by showing that for each candidate strategy τ "trying" to dominate a bad strategy σ, 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 (a player never broadcasts) 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.

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, points out that the suggested 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. Note that 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

On the backward induction process. Previously suggested rational secret sharing schemes make use of cryptographic primitives (e.g., see [1, 12]). However, unlike standard cryptographic protocols, those schemes may run for an exponential number of rounds (at least with an exponentially small probability). The key problem is that there is a (large) bound rounds b, 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 from 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, see formal definition in Appendix B. The notion of everlasting equilibrium is related to known Game Theoretic concepts, see discussion in Remark B.9.

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.

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 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. $\boldsymbol{\sigma}$ 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. Formal definition can be found in Appendix C.

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 $f(\mathbf{x})$) when outcome o is reached, if according to o the player quits and outputs the right value. Let o and o' be two possible outcomes of the game, and let $\mathsf{retrieve}(o)$ be the set of players retrieving the value when o is reached:

1. $u_i(o) > u_i(o')$ whenever $i \in \mathsf{retrieve}(o)$ and $i \notin \mathsf{retrieve}(o')$ (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 computing equilibria protocols that 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. Clearly, players are never worse off quitting in such cases, 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: If a player runs a non-linger avoiding strategy, then the "linger avoiding version" of his strategy is another best response, and thus no strict equilibria can be found. Therefore, we use the weaker notion of a strict Nash equilibrium with respect to linger avoiding strategies, and only require each player's strategy to be a best response that is strictly better than any *linger avoiding* strategy that acts differently on the equilibrium path (for a formal definition see Appendix B).

4 Impossibility Results

We show that for *every* non-constant function f with a *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. Formal definition and complete proofs of all the claims made in this section can be found in Appendix D.

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 transcript of length t of a protocol $\boldsymbol{\sigma}$ is a sequence of messages that players may broadcast when running $\boldsymbol{\sigma}$ for t rounds. We view the transcripts of $\boldsymbol{\sigma}$ 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 transcript $\mathbf{m}$ given input x_i, if given player i's view there is only one possible value for $f(\mathbf{x})$. That is, 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* 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 4.1. *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 linger avoiding, Nash equilibrium protocol for Γ_f that computes f and does not have a revelation point.*

We next sketch the proof of the above theorem, a complete proof can be found in Appendix D. 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 after the child $\mathbf{m}$ of $\mathbf{p}$, but do not learn 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 decide to act according to $\mathbf{m}$, it is also possible that player 1 will learn the secret alone. Since we assume 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.

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

Corollary 4.2. *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 **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:

Corollary 4.3. *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.*

For completeness, we state the following easy claim regarding the NSCG model:

Claim 4.4. *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 4.1, 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 Secret Sharing 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 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.

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 3 in Appendix E.

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.[4]

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.

[4] For example, this can be done using the following method (see [19, 15]): 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 $c_i = b_i \cdot x + s_i$.

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

Note 5.1. 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 5.2 (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 assume that players prefer to learn the secret without the others):

First, β should be chosen to be small enough. Clearly, 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 utility functions, and are calculated in Appendix E. The theorem's proof can also be found the appendix, and it 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 5.2. *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 $2 \leq m \leq n$, the scheme described above is a simultaneous* ***strict*** *rational m-out-of-n secret sharing scheme for Y with respect to linger avoiding strategies. It has expected running time $O(\frac{1}{\beta^2})$, and expected share size $O(\frac{1}{\beta} \log \frac{1}{\beta})$.*

6 An ε-Rational Secret Sharing 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, and 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. A formal description of the dealer's algorithm, as well as some additional remarks, can be found in Appendix F.

Claim 4.4 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. Indeed, 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 6.1. *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 $O\left(\frac{n \lg n}{\beta}(\log \frac{1}{\beta} + \log \frac{U_{max}}{\varepsilon})\right)$.*

The proof is similar to that of Theorem 5.2.

Note 6.2. 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.

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 [9].

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] O. Goldreich. Foundations of Cryptography, volume 2: Basic Applications, *Cambridge University Press*, 2004.

[5] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. *In Proceedings of the 19th ACM Symposium on Theory of Computing (STOC)*, pages 218-229, 1987.

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

[7] 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.

[8] 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.

[9] G. Kol and M. Naor. Cryptography and Game Theory: Designing Protocols for Exchanging Information. *To appear in the Proceedings of the 5th Theory of Cryptography Conference (TCC),* 2008.

[10] 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.

[11] 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.

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

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

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

[15] 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.

[16] L. Samuelson. Dominated Strategies and Common Knowledge. *Games and Economic Behavior*, volume 4, issue 2, pages 284-313, 1992.

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

[18] D. Stahl. Lexicographic Rationalizability and Iterated Admissibility, *Economic Letters*, volume 47, pages 155-159, 1995.

[19] 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.

A Solution Concept

A.1 On Iterated Admissibility

In this section we suggest further arguments as to why iterated admissibility should not be used to distinguish "good" information exchange protocols from "bad" ones. We first review the definition of iterated admissible strategies, as it appears in [7]:

Definition A.1 (Iterated Admissible Strategies). *Let S_i denote a set of (randomized) strategies for player i, $\mathbf{S}_{-i} = S_1 \times ... \times S_{i-1} \times S_{i+1} \times ... \times S_n$, and u_i the utility function of player i. A strategy $\sigma_i \in S_i$ is* weakly dominated *by a strategy $\sigma'_i \in S_i$ with respect to $\mathbf{S}_{-i}$ if:*

- *There exists a $\boldsymbol{\sigma}_{-i} \in \mathbf{S}_{-i}$ such that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) < u_i(\sigma'_i, \boldsymbol{\sigma}_{-i})$ and*

- *For all $\boldsymbol{\sigma}_{-i} \in \mathbf{S}_{-i}$ it holds that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) \leq u_i(\sigma_i', \boldsymbol{\sigma}_{-i})$*

Strategy σ_i is weakly dominated with respect to $\mathbf{S}_{-i}$ if there exists a $\sigma_i' \in S_i$ such that σ_i is weakly dominated by σ_i' with respect to $\mathbf{S}_{-i}$.

Let $\mathsf{DOM}_i(S_1 \times ... \times S_n)$ denote the set of strategies in S_i that are weakly dominated with respect to $\mathbf{S}_{-i}$. Let S_i^0 denote the initial set of allowable strategies for player i. For all $k \geq 1$, define S_i^k inductively as $S_i^k = S_i^{k-1} \backslash \mathsf{DOM}_i(S_1^{k-1} \times ... \times S_n^{k-1})$. $S_i^\infty = \bigcap_k S_i^k$ is the set of (exhaustive) iterated admissible strategies *for player i.*

One argument against using iterated admissibility is Game Theoretic: Samuelson [16] and Stahl [18] showed that the assumption that rational players follow only iterated admissible strategies poses extraneous and hard-to-justify restrictions on players' beliefs. Roughly speaking, Samuelson showed that iterated admissibility is not equivalent to the natural notion of common knowledge of admissibility (everyone knows that everyone avoids playing weakly dominated strategies; everyone knows that everyone knows it; and so on). Instead, it was shown by Stahl that a strategy survives elimination if it is a best response to a belief that one strategy is infinitely less likely than another if the former is eliminated at an earlier round than the latter.

In addition, since the order in which strategies are eliminated can affect the outcome, Halpern and Teague's choice of exhaustive elimination, i.e, removing all weakly dominated strategies in every round, is not the only option.

Another argument is that the `talk-once` strategy, and many more "bad" strategies, are actually iterated admissible in the `one-time-shares` model described below. The model is intended to match the one used by Halpern and Teague, though there are many details they do not make explicit.

The `one-time-shares` model. The model consists of an honest dealer that chooses a secret out of a finite set, and n players that try to learn the secret. 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 a sequence of iterations, each iteration may consist of multiple communication rounds. At the end of an iteration, the protocol always proceeds to the next one, thus the underlying game is infinite.

At the beginning of each iteration, 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 the 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. That is, the deviating behavior of players is limited to refusal to cooperate, ignoring the case of a player reporting an incorrect share. 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.

Remark A.2. 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, our claim 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. (If m or more players broadcast, all players learn. When $m-1$ players broadcast, the rest of the players learn, and have no incentive to continue participating in the game. The remaining $m-1$ players have no way of learning the secret by themselves).

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. Note that `talk-once` is such a strategy.

Theorem A.3. *In the* `one-time-shares` *model for rational m-out-of-n secret sharing ($2<m<n$), for every $i \in N$ it holds that $A_i \subseteq S_i^{\infty}$. In particular,* $\texttt{talk-once}_i \in S_i^{\infty}$.

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 is $\tau_i \in S_i$ that weakly dominates σ_i. In particular, there is $\boldsymbol{\tau}_{-i} \in \mathbf{S}_{-i}$ 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} are different.

Let $h = (\mathbf{b}_1, ..., \mathbf{b}_t)$ be a boolean history, 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 holds 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}, we get $u_i(R^{\tau_i}) = u_i(R^{\sigma_i})$, and thus 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 history i learns the secret and some of the others do not, whereas if the history is not h, all players learn.

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

If (3) holds: assume that the other players follow $\boldsymbol{\sigma}^{h,+}_{-i}$. 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, $u_i(\sigma_i, \boldsymbol{\sigma}^{h,+}_{-i}) > u_i(\tau_i, \boldsymbol{\sigma}^{h,+}_{-i})$, and $\boldsymbol{\sigma}^{h,+}_{-i}$ "saves" σ_i.

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

Remark A.4. It might seem somewhat artificial to view `talk-once` as strategy in an infinite game, since it only has one "actual" stage. However, it is not hard to see that the following version of `talk-once` is iterated admissible as well, and may have any number of "actual" stages:

talk-once*$_i$:

In round s:

- *If* $s = 0$ *or* $[s > 0$ *and* $\bigoplus_{j \in N} a^{s-1}_j \neq 0]$: choose a bit at random (a^s_i) and broadcast it.
- *Else*: broadcast your current share. From now on, keep silent.

B Game Theoretic Definitions

B.1 Extensive Form Game with Imperfect Information and Behavioral Strategies

Information exchange protocols can be viewed as behavioral strategies in an extensive form game with imperfect information. Extensive form games model multi-staged interactions between players. In extensive form games with imperfect information we allow players to be imperfectly informed about past events when taking actions (e.g., in our settings, players are uninformed of the inputs assigned to the others). In addition, we permit exogenous uncertainty, that is, some moves may be made by "chance" (e.g., in our settings, the role of "chance" is confined to choosing the players' inputs at the beginning of the game). A behavioral strategy is a randomized algorithm, determining the player's action in every situation.

Definition B.1 (Extensive Form Game with Impefrect Information). *An* extensive form game with imperfect information *is a tuple* $\langle N, H, P, f_c, (\mathcal{I}_i)_{i \in N}, (u_i)_{i \in N} \rangle$ *where:*

- N - **Players**: *a finite set of players denoted* $1, ..., n$.
- H - **Histories**: *a set of sequences (finite or infinite) that satisfies the following properties:*
  - *The empty sequence* ϕ *is in* H.
  - *If* $(a_1, ..., a_K) \in H$ *(where* K *might be infinite) and* $L < K$ *then* $(a_1, ..., a_L) \in H$.
  - *If an infinite sequence* $(a_1, a_2, ...)$ *satisfies* $(a_1, ..., a_L) \in H$ *for every positive integer* L *then* $(a_1, a_2, ...) \in H$.

  Each member of H *is a* history*; each component of a history is an* action *taken by a player. A history* $(a_1, ..., a_K) \in H$ *is a* terminal history *(or an* outcome*) if it is infinite or if there is no* a_{K+1} *such that* $(a_1, ..., a_K, a_{K+1}) \in H$. *The set of actions available after a non-terminal history* h *is denoted* $A(h) = \{a \mid (h, a) \in H\}$, *and the set of terminal histories is denoted* Z.
- P - **Next player**: *a function* $P : (H \setminus Z) \to N \cup \{c\}$ *for which* $P(h)$ *is the player who takes an action after the history* h. *If* $P(h) = c$ *then "chance" determines the action taken after history* h.
- f_c - **Chance's distributions**: *a function that associates with every history* h *for which* $P(h) = c$, *a probability measure* $f_c(\cdot, h)$ *on* $A(h)$ *(i.e.,* $f_c(a|h)$ *determines the probability that action* a *occurs after the history* h*). The probability measures are independent.*

- $\mathcal{I}_i$ - **Information partition** for player i: *for each player $i \in N$, the set $\mathcal{I}_i$ is a partition of $\{h \in H \mid P(h) = i\}$ with the property that $A(h) = A(h')$ whenever h and h' are in the same member of the partition. A set $I_i \in \mathcal{I}_i$ is an* information set *of player i.*
- u_i - **Utility (payoff) function** for player i: *a function $u_i : Z \to \mathbb{R}$ determining player i's gain for every outcome.*

Note B.2. Using the above definition we can model simultaneous moves: Players choose actions one after the other, but no player gets informed of the previous actions selected by the other players before selecting his.

Definition B.3 (Behavioral Strategy)**.** *Let Γ be an extensive form game with imperfect information. A* behavioral strategy *for player i in Γ is a collection $(\gamma(I_i))_{I_i \in \mathcal{I}_i}$ of independent probability measures, where $\gamma(I_i)$ is a probability measure over the set of possible actions for every history in I_i.*

Note that in games with perfect recall (information learned is not forgotten) such as ours, behavioral strategies are equivalent to mixed strategies (probability measures over deterministic strategies).

B.2 Behavioral Strategies and Rationality Concepts

In this section we formalize the Game Theoretic stability notions used in the paper. We start off by reviewing the standard concepts. We call a vector of players strategies a *strategy profile*, and use the following notations: $\boldsymbol{\alpha}_{-i} = (\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_n)$, $(\boldsymbol{\alpha}_{-i}, \alpha'_i) = (\alpha_1, ..., \alpha_{i-1}, \alpha'_i, \alpha_{i+1}, ...\alpha_n)$, and $u_i(\boldsymbol{\sigma}) = \mathbf{E}_{o \sim \mathcal{O}(\boldsymbol{\sigma})}[u_i(o)]$ where $\mathcal{O}(\boldsymbol{\sigma})$ denotes the probability distribution over outcomes induced by the protocol $\boldsymbol{\sigma}$.

Definition B.4 (Nash Equilibrium)**.** *A behavioral strategy profile $\boldsymbol{\sigma}$ for the game Γ is said to be a* Nash equilibrium *if for every $i \in N$ and any behavioral strategy σ'_i, it holds that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) \geq u_i(\sigma'_i, \boldsymbol{\sigma}_{-i})$.*

Definition B.5 (ε-Nash Equilibrium)**.** *A behavioral strategy profile $\boldsymbol{\sigma}$ for the game Γ is said to be an* ε-Nash equilibrium *if for every $i \in N$ and any behavioral strategy σ'_i, it holds that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) + \varepsilon \geq u_i(\sigma'_i, \boldsymbol{\sigma}_{-i})$.*

Definition B.6 (Strict Equilibrium with respect to **A**)**.** *Let $\boldsymbol{\sigma}$ be a behavioral strategy profile for the game Γ, and let A_i be a subset of behavioral strategies for player i. $\boldsymbol{\sigma}$ is said to be a* strict Nash equilibrium *with respect to $\mathbf{A} = (A_1, ..., A_n)$ if for every $i \in N$ and any $\sigma'_i \in A_i$ that assigns a different action to some information set of player i that can be reached when following $\boldsymbol{\sigma}$, it holds that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) > u_i(\sigma'_i, \boldsymbol{\sigma}_{-i})$.*

Note B.7. The standard definition of a strict equilibrium is stronger than ours. It demands that σ_i is a better response to $\boldsymbol{\sigma}_{-i}$ than any σ'_i that differs from it on *some* information set, whereas we only require it to be strictly better than any strategy in A_i that acts differently on an information set *that can be reached.*

Next, we formalize the notion of everlasting equilibrium suggested in Section 2.2.

Definition B.8 (Everlasting, Strict Everlasting with respect to **A**, and ε-Everlasting Equilibrium)**.** *Let Γ be a game, I be an information set of Γ, and μ be a probability measure on the set of histories in I. We interpret μ as the probability that current player assigns to the history $h \in I$, conditional on I being reached. For a behavioral strategy profile $\boldsymbol{\sigma}$ we define $\mathcal{O}(\boldsymbol{\sigma}, \mu \mid I)$ to be the distribution over terminal histories determined by $\boldsymbol{\sigma}$ and μ, conditional on I being reached, as follows. Let $h* = (a_1, ..., a_K)$ be a terminal history. Then:*

- *If there is no subhistory of $h*$ in I (i.e., the information set that the game has reached rules out $h*$), then $\mathcal{O}(\boldsymbol{\sigma}, \mu \mid I)(h*) = 0$.*
- *If the subhistory $h = (a_1, ..., a_L)$ of $h*$ is in I, where $L < K$, then $\mathcal{O}(\boldsymbol{\sigma}, \mu \mid I)(h*) = \mu(h) \cdot \prod_{k=L}^{K-1} \sigma_{P(a_1,...,a_k)}(a_1, ..., a_k)(a_{k+1})$, where $\sigma_{P(a_1,...,a_k)}(a_1, ..., a_k)(a_{k+1})$ denotes the probability that the player taking a move after history $(a_1, ..., a_k)$ chooses the action a_{k+1}.*

$\boldsymbol{\sigma}$ is said to be an everlasting (strict everlasting *with respect to* $\mathbf{A}$, ε-everlasting) equilibrium *if for every player $i \in N$, every information set $I_i \in \mathcal{I}_i$ that can be reached by $\boldsymbol{\sigma}$, and any behavioral strategy σ_i' for player i, it holds that $u_i\left(\mathcal{O}\left((\boldsymbol{\sigma}_{-i}, \sigma_i), \mu_{I_i} \mid I_i\right)\right) \geq u_i\left(\mathcal{O}\left((\boldsymbol{\sigma}_{-i}, \sigma_i'), \mu_{I_i} \mid I_i\right)\right)$, where $u_i(\mathcal{O}) = \mathbf{E}_{o \sim \mathcal{O}}[u_i(o)]$ and μ_{I_i} is the distribution over histories in I_i induced by $\boldsymbol{\sigma}$.*

*For a **strict** everlasting equilibrium replace "$\geq$" by "$>$", and for **ε**-everlasting add "$+\varepsilon$" to the the LHS.*

Remark B.9. The concept of everlasting equilibrium resembles the Game Theoretic notion of a subgame perfect equilibrium (or sequential equilibrium), but is strictly 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 *on the equilibrium path,* and it eliminates "non-credible promises" (e.g., players' non-credible promises to cooperate in the last round).

C Our Settings

C.1 Settings for Rational Secret Sharing

Definition C.1 (Rational, Strict Rational with respect to $\mathbf{A}$, and ε-Rational m-out-of-n secret sharing scheme)**.** *Let Y be a finite set of secrets, $\mathcal{D}_Y$ a distribution over Y (wlog assume that every $y \in Y$ has a positive probability), and $(u_i)_{i \in N}$ the given utility functions. Let $\texttt{dealer} : Y \longmapsto \mathbf{X}$, ($\mathbf{X} \subseteq \times_{i \in N} X_i$), be a probabilistic mapping that associate shares to secrets, with a reconstruction function $f : \mathbf{X} \to Y$. Let $\boldsymbol{\sigma}$ be a protocol for $\Gamma_f^{\mathcal{D},(u_i)_{i \in N}}$.*

The pair $(\texttt{dealer}, \boldsymbol{\sigma})$ *is a* simultaneous rational (strict rational *with respect to* $\mathbf{A}$, ε-rational) m-out-of-n secret sharing scheme *for Y with respect to $\mathcal{D}_Y$ and $(u_i)_{i \in N}$, if:*

- ***The secret can be rationally reconstructed using $\boldsymbol{\sigma}$ by any subset C of m or more players:** there is a secret reconstructing function $f_C : \mathbf{X}_C \to Y$, ($\mathbf{X}_C \subseteq \times_{i \in C} X_i$), such that the strategies prescribed to players in C are an **everlasting (strict everlasting** with respect to **$\mathbf{A}$, ε-everlasting) equilibrium** that computes f_C in the corresponding computing game Γ_{f_C}(the distribution over inputs in $\mathbf{X}_C$ is determined by $\mathcal{D}_Y$ and $\texttt{dealer}$, and the utility functions are the ones induced by $(u_i)_{i \in C}$ when players not in C are assumed to never broadcast).*
- ***No subset C of less than m players can reveal any partial information about the secret (in the information theoretic sense) before the game begins**: the distribution over inputs given any shares of players in C is identical to the original distribution $\mathcal{D}_Y$.*

Non-simultaneous rational schemes *are defined similarly, by replacing $\Gamma_f^{\mathcal{D},(u_i)_{i \in N}}$* with $\bar{\Gamma}_f^{\mathcal{D},(u_i)_{i \in N}}$.

D Impossibility Results

In this section we proof the impossibility results of Section 4.

Remark D.1. An impossibility result in the same spirit was offered by Halpern and Teague ([7], Corollary 3.1). They claim that (under some harsh restrictions) there is no iterated admissible Nash equilibrium protocol for 2-*out-of*-2 secret sharing in a model with simultaneous broadcast and private channels. We remove those restrictions: We offer results for more than 2 players, do not assume the iterated admissibility of the protocols, Shamir shares as the player's inputs, unforgeable signatures, or that the inputs are atomic and cannot be subdivided. In addition, as pointed out by Abraham et al., Halpern and Teague's proof seems to be problematic (see the discussion after Definition 2 in Section 4 of [1]).

Note that we assume that the only mean of communication between players is a broadcast channel, whereas in Halpern and Teague's setting communication via (simultaneous) private channels is also allowed. However, in the 2 players case addressed by their impossibility result, the assumption of private channels is equivalent to that of a broadcast channel. Therefore, our results do not pose any new constraints.

D.1 Transcripts Trees and Revelation Points

We formally define the terms described in Sections 4.1 and 3.4:

Definition D.2 (Run). *Let $\boldsymbol{\sigma}$ be a protocol for the SCG Γ_f. 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.*

Definition D.3 (Transcript, Explains). *Let $\boldsymbol{\sigma}$ be a protocol for the SCG Γ_f. 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:

Definition D.4 (Transcripts Tree). *Let $\boldsymbol{\sigma}$ be a protocol for the SCG Γ_f. The* transcripts tree *of $\boldsymbol{\sigma}$ is a tree whose vertices are the elements of $M(\boldsymbol{\sigma})$. The tree's root is the empty history, and $\mathbf{m} = (\mathbf{m}_1, ..., \mathbf{m}_t)$ is the parent of $\mathbf{m}' = (\mathbf{m}'_1, ..., \mathbf{m}'_{t+1})$ if for every $s \leq t$, it holds that $\mathbf{m}_s = \mathbf{m}'_s$.*

Definition D.5 (Learns / Knows). *Let $f : \mathbf{X} \to Y$ be a function, $\boldsymbol{\sigma}$ a protocol for the SCG Γ_f, $R = (\mathbf{x}, \mathbf{r})$ a run of $\boldsymbol{\sigma}$, and $\mathbf{m}$ a transcript of $\boldsymbol{\sigma}$. The following are phrases and their meanings:*

- Player i learns (knows) $f(\mathbf{x})$ after $\mathbf{m}$ given x_i: *There exists $y \in Y$ such that for every run $R' = (\mathbf{x}', \mathbf{r}')$ of $\boldsymbol{\sigma}$ with $x'_i = x_i$ and $\mathsf{m}(R', t) = \mathbf{m}$, it holds that $f(\mathbf{x}') = y$.*
- Player i learns (knows) $f(\mathbf{x})$ after round t of R: *i learns after $\mathbf{m} = \mathsf{m}(R, t)$ given x_i.*
- Player i learns (knows) $f(\mathbf{x})$ during round t of R: *Player i does not know $f(\mathbf{x})$ after round $t - 1$ of R, but does know it after round t.*

Definition D.6 (Revelation Point). *Let $\boldsymbol{\sigma}$ be a protocol for the SCG Γ_f, and $\mathbf{m}$ a finite transcript of $\boldsymbol{\sigma}$. $\mathbf{m}$ is a* revelation point *of $\boldsymbol{\sigma}$ if:*

- *There exists an input $\mathbf{x} \in \mathbf{X}$ and a player $i \in N$ such that i does not know $f(\mathbf{x})$ after $\mathbf{m}$* given *x_i.*
- *For every input $\mathbf{x} \in \mathbf{X}$ and any player $i \in N$, i knows $f(\mathbf{x})$ after any child of $\mathbf{m}$ given x_i.*

Definition D.7 (Linger Avoiding). *A strategy σ_i for player i in Γ_f is* linger avoiding *if for every joint strategy of the other players $\boldsymbol{\sigma}_{-i}$, and every run R of the protocol $(\sigma_i, \boldsymbol{\sigma}_{-i})$, if player i learns $f(\mathbf{x})$ during round t of R, he quits in round $t + 1$. A protocol $\boldsymbol{\sigma}$ for Γ_f is linger avoiding if the strategy σ_i is linger avoiding for every $i \in N$.*

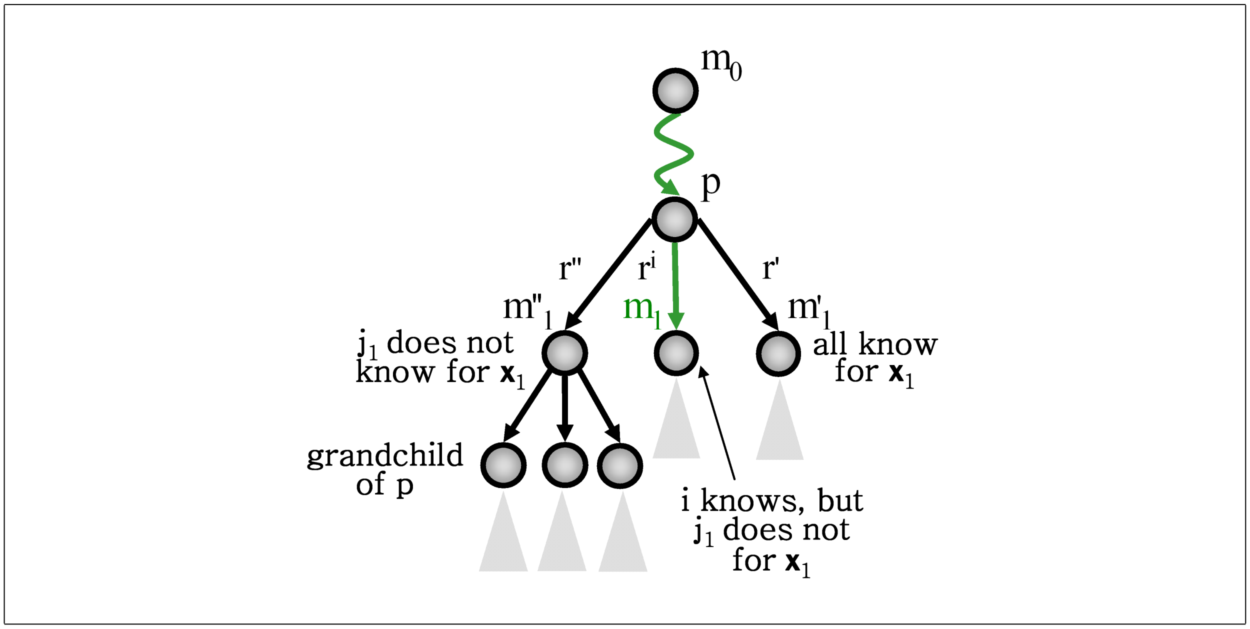


Figure 1: The transcripts tree for the protocol $\boldsymbol{\sigma}$, and the selection of the next vertex on branch leading to the revelation point.

D.2 Impossibility Results for Rational MPC

Theorem (4.1 restated)**.** *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, Nash equilibrium protocol for Γ_f that computes f and does not have a revelation point.*

Proof Let $\boldsymbol{\sigma}$ be such a protocol. For $\mathbf{m} \in M(\boldsymbol{\sigma})$ define:

$$C_{\mathbf{m}} = \{(i, \mathbf{x}) \mid i \in N, \mathbf{x} \in \mathbf{X}, \text{ and } i \text{ does not know } f(\mathbf{x}) \text{ after } \mathbf{m} \text{ given } x_i\}$$

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. See Figure 1 for a sketch of the tree's structure.

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}.

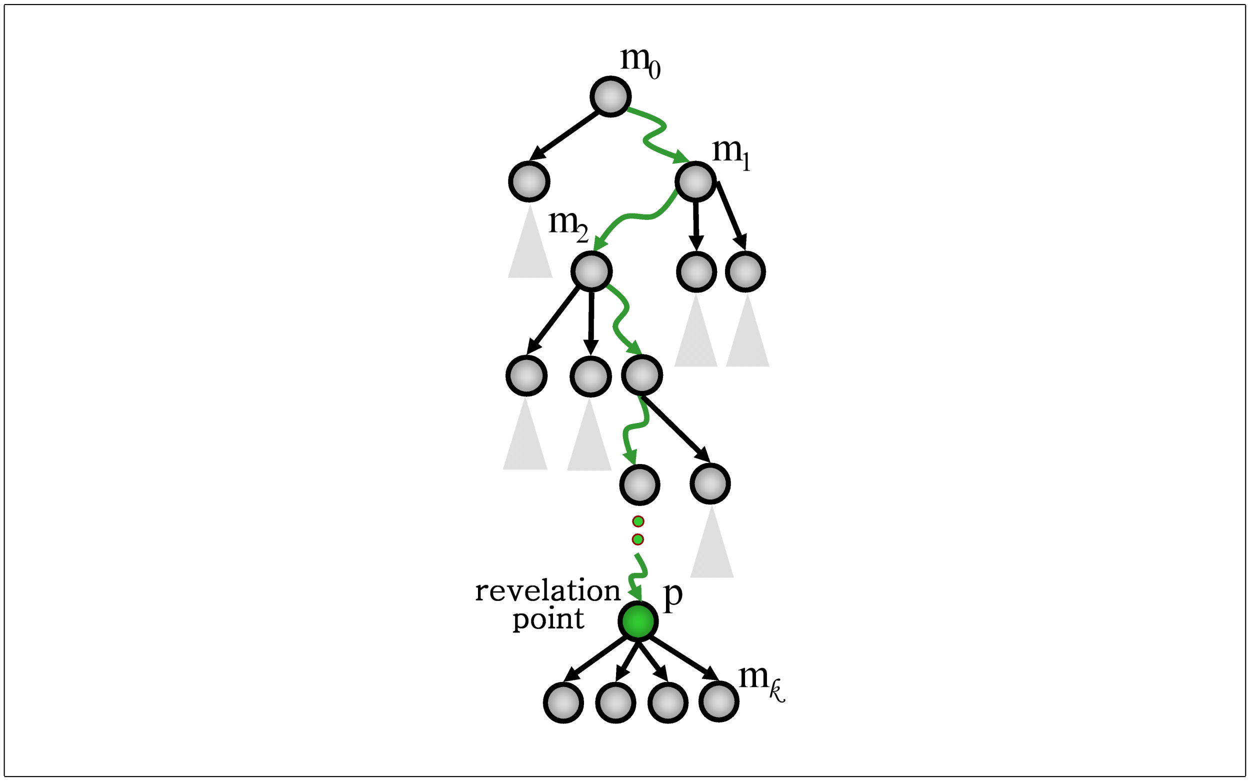


Figure 2: The transcripts tree for the protocol $\boldsymbol{\sigma}$, and the branch leading to its revelation point built by the proof.

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 $(i, \mathbf{x}_1) \in C_{\mathbf{m}_0} \setminus C_{\mathbf{m}_1}$.

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. See illustration in Figure 2.

■

Remark D.8. Theorem 4.1 does not imply that there is an efficient algorithm for finding revelation points, or that there even exists such an algorithm. However, since a revelation point $\mathbf{m}_{rev}$ does exist, player i may prefer to deviate from the strategy σ_i and follow $\sigma_i^{\mathbf{m}_{rev}}$ (described below).

$\underline{\sigma_i^{\mathbf{m}_{rev}}(x, \mathbf{m}, r)}$:

- *If* $\mathbf{m} = \mathbf{m}_{rev}$: keep silent.
- *If* $\mathbf{m}$ *is a child of* $\mathbf{m}_{rev}$: quit and output the value you have just learned.
- *Else*: run $\sigma_i(x, \mathbf{m}, r)$.

Corollary (**4.2** restated)**.** *Let f be a non-constant function with a 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.*

Proof Assume for contradiction that $\boldsymbol{\sigma}$ is such a protocol. Since $\boldsymbol{\sigma}$ is a Nash equilibrium protocol that computes f, any strategy that allows player i to learn the value almost always is a best response to $\boldsymbol{\sigma}_{-i}$ (due to Properties 2 and 3 of strictly competitive utility functions). If $\boldsymbol{\sigma}$ is not linger avoiding, then the linger avoiding version of σ_i (player i follows σ_i with one exception: he quits immediately after learning $f(\mathbf{x})$) is another best response to $\boldsymbol{\sigma}_{-i}$. Therefore, $\boldsymbol{\sigma}$ must be linger avoiding, and by Theorem 4.1, it has a revelation point $\mathbf{m}_{rev}$.

Consider the strategy $\sigma_i^{\mathbf{m}_{rev}}$ described in Remark D.8 (player i broadcasts messages according to σ_i, with the exception that after $\mathbf{m}_{rev}$ he keeps silent). The strategy $\sigma_i^{\mathbf{m}_{rev}}$ is a best response to $\boldsymbol{\sigma}_{-i}$ since the strategies σ_i and $\sigma_i^{\mathbf{m}_{rev}}$ only differ after the transcript $\mathbf{m}_{rev}$ is reached, and when following $\sigma_i^{\mathbf{m}_{rev}}$, player i retrieves the value after $\mathbf{m}_{rev}$.

Since $\sigma_i^{\mathbf{m}_{rev}}$ is a linger avoiding strategy (as σ_i is such), and $\boldsymbol{\sigma}$ is a strict Nash equilibrium with respect to linger avoiding strategies, σ_i and $\sigma_i^{\mathbf{m}_{rev}}$ must act the same for every history reachable when following $\boldsymbol{\sigma}$. However, if every player $i \in N$ follows $\sigma_i^{\mathbf{m}_{rev}}$, then all players are keeping silent after $\mathbf{m}_{rev}$ for every set of inputs. This behavior leads to $\mathbf{m}_{rev}$ having a single child $\mathbf{m}$. Since players learn $f(\mathbf{x})$ after $\mathbf{m}$, they must have already learned it after $\mathbf{m}_{rev}$. This contradicts our assumption that $\mathbf{m}_{rev}$ is a revelation point. ■

Corollary (**4.3** restated)**.** *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.*

Proof Assume for contradiction that that $\boldsymbol{\sigma}$ is such a protocol. We first show that both players must learn the value during the same round. Assume that this is not the case, and let $R = (\mathbf{x}_0 = (x_1^0, x_2^0), \mathbf{r})$ be a run of $\boldsymbol{\sigma}$ for which, wlog, player 1 learns the value after round t, but player 2 does not. Since player 2 is still unsure of the value after round t, there is at least one input other than player 1's real input that seems possible to player 2, but results in a different value for f. By following the strategy $\sigma_1^{x_1^0}$ (described below) that guesses one of the other inputs and pretends to be holding it, player 1 can prevent player 2 from learning $f(\mathbf{x})$ with a positive probability. Properties 2 and 3 of strictly competitive utility functions imply that player 1 is better off following $\sigma_1^{x_1^0}$.

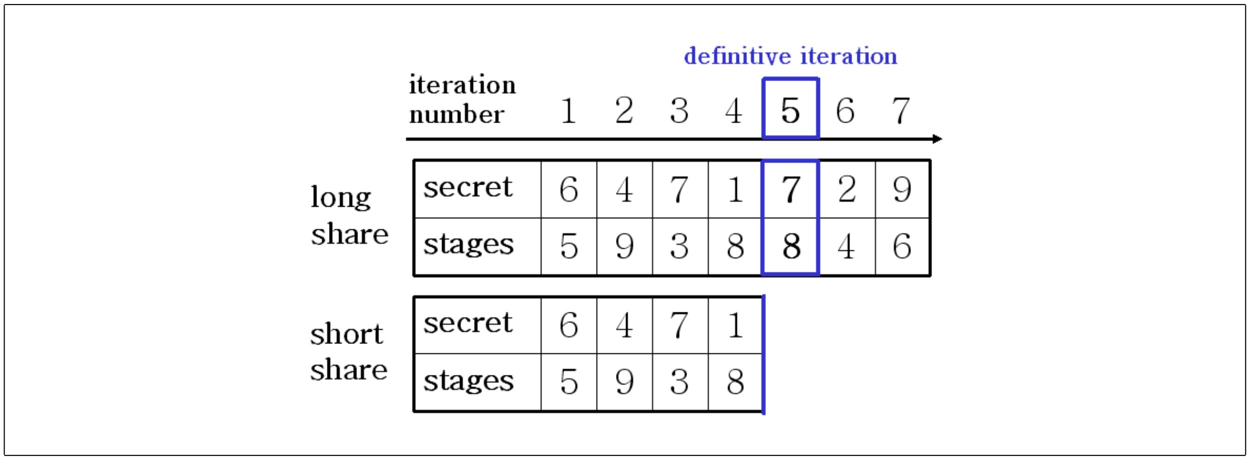
$\underline{\sigma_1^{x_1^0}(x, \mathbf{m}, r)}$:

- *If* $\mathbf{m} = \mathsf{m}(R, t)$ *and* $x = x_1^0$: randomly select an input $\mathbf{x}' = (x_1', x_2') \in \mathbf{X}$ with $x_1' \neq x$, and random tapes $\mathbf{r}' = (r_1', r_2')$, such that the run $R = (\mathbf{x}', \mathbf{r}')$ of $\boldsymbol{\sigma}$ explains $\mathbf{m}$. From now on, use the value x_1' instead of x, with the exception that when quitting, the secret guessed is the one learned after $\mathsf{m}(R, t)$.
- *Else:* run $\sigma_i(x, \mathbf{m}, r)$

Now, since players learn the value together, we may repeat the proof of Theorem 4.1 without the need of requiring that $\boldsymbol{\sigma}$ is linger avoiding, and show the existence of a revelation point: Since player i learns after round t of R^i, the other player must learn too, but this contradicts the assumption that some players do not learn. Again, when the revelation point is reached player 1 is better off deviating from σ_i and pretending to be holding a fictitious input. ■

Claim (**4.4** restated)**.** *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.*

Proof As argued in the proof of Corollary 4.3, if one of the players learns before the other, he may pretend to be holding a different input and fool the other player with a positive probability (since $\mathbf{X}$ is countable we can assume, that every $\mathbf{x} \in \mathbf{X}$ has a positive probability according to $\mathcal{D}$). Since it is not possible for both players to learn together in the NSBC model, the claim holds. ■



	iteration number	1	2	3	4	5	6	7
long share	secret	6	4	7	1	7	2	9
	stages	5	9	3	8	8	4	6
short share	secret	6	4	7	1			
	stages	5	9	3	8			

Figure 3: Example of possible shares assigned by the dealer's algorithm.
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 $d = 3$.

E A Strict Rational Secret Sharing Scheme with Unbounded Shares

Formal description of the scheme. An example for (partial) shares distributed by the dealer's algorithm is given in Figure 3. Formal description of the dealer's and players' algorithms described in Section 5 can be found in Figures 4 and 5.

Some additional notes. The following are some additional notes regarding the suggested scheme:

Note E.1. 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.

Note E.2. Although the boolean indicator added in the general case allows a set of long player to identify the definitive iteration by themselves, it does not replace the need of assigning a shorter share to one of the players: if all shares are of the same size, the length of the game is known to all the players, allowing them deviate in the last iteration.

The values of β_0 and c_0, and the proof of Theorem 5.2. We next calculate the values β_0 and c_0, and then prove Theorem 5.2. As discussed in Section 5, we cannot expect a player to participate in a sharing scheme if he can a-priori guess the secret with a sufficiently high probability.

More formally, let U_i and U_i^+ be the minimal and maximal payoffs of player i when he retrieves the secret, and let U_i^- be the maximal payoff of i in case he does not retrieve. Denote by α player i's chance of guessing the secret at the beginning of the game, given his share and the initial distribution $\mathcal{D}$. If player i does not participate in the protocol, he guesses the right secret with probability at most α and gets at most U_i^+. However, with probability $1 - \alpha$ he guesses a wrong secret, and gets at most U_i^-. By participating in the game, player i ensures a payoff of at least U_i. Therefore, if the following inequality is satisfied, player i has an incentive to participate in a sharing scheme.

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 possible 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* possible 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 value indicating whether this iteration is definitive.
  - **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 $1 - \beta$ (can be achieved with tag and hash of size $\log \frac{1}{\beta}$).

  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 4: The dealer's shares assignment algorithm

```
Player_i(share)
Set secret_revealed ← FALSE and cheater_detected ← FALSE.
Repeat until secret_revealed = TRUE or cheater_detected = TRUE
  • If your share ended:
      – Keep silent.
      – If someone has broadcasted, secret_revealed ← 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 cheater_detected ← TRUE.
      – If this is the last stage:
          ∗ Broadcast the 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),
            cheater_detected ← TRUE.
          ∗ If all but a single player broadcasted, or if the reconstructed indicator shows that
            the iteration is definitive, secret_revealed ← 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 5: Player i's reconstruction protocol

$$\begin{aligned} \alpha U_i^+ + (1-\alpha)U_i^- &< U_i \\ \alpha(U_i^+ - U_i^-) &< U_i - U_i^- \\ \alpha &< \frac{U_i - U_i^-}{U_i^+ - U_i^-} \end{aligned}$$

Denote $c_i = \frac{U_i - U_i^-}{U_i^+ - U_i^-}$ and $c_0 = \min_{i \in N}\{c_i\}$ (since the payoff functions are learning preferring, it hold that $U_i^- < U_i \leq U_i^+$, and thus $c_i > 0$). Since $\alpha \geq \mathcal{D}(b)$, we at least need to require $\mathcal{D}(b) < c_0$. The following proof shows that it suffices to set $\beta_0 = \min_{i \in N}\left\{\frac{c_i - \mathcal{D}(b)}{(c_i - \mathcal{D}(b)) + 2z \cdot n + 1}\right\}$ where $z = |Y|$ and Y is the set of secrets (since $c_i > \mathcal{D}(b)$, it holds that $\beta_0 > 0$).

Theorem (**5.2** restated)**.** *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 $2 \leq m \leq n$, the scheme described above is a simultaneous* ***strict*** *rational m-out-of-n secret sharing scheme for Y with respect to linger avoiding strategies. It has expected running time $O(\frac{1}{\beta^2})$, and expected share size $O(\frac{1}{\beta}\log\frac{1}{\beta})$.*

Proof We need to show:

1. No group of less than m players is able to learn anything about the secret before the game begins.
2. Every subset of at least m players following their prescribed strategies reconstructs the secret.
3. The expected running time of the reconstruction protocol and expected share size are as claimed.
4. The strategies prescribed to every subset of at least m players are strict best responses after any history h.

To show (1), recall that all values appearing in players' shares, aside from the masked secrets, are chosen independently of the real secret, and thus cannot be used to extract any information about it. Since the masks are randomly chosen and shared using an m-out-of-n scheme, a set of less than m players cannot learn anything about the masks. Hence, the masked secrets cannot be used to gain knowledge of the secret either.

To show (2) and (3) we claim that every subset of m or more players following the protocol learns the secret after the definitive iteration ℓ. Since the definitive iteration and the number of stages in any iteration are chosen according to $\mathcal{G}(\beta)$, the expected running time is $O(\frac{1}{\beta^2})$. The expected share size is $O(\frac{1}{\beta}\log\frac{1}{\beta})$ due to the fact that a share can either be of length $\ell - 1$ or $L = \ell + d - 1$ for d chosen according to $\mathcal{G}(\beta)$, and the size of each cell is $O(\log\frac{1}{\beta})$.

It remains to prove (4), that is, we need to show that as long as no deviation was detected, every player that does not know the secret is strictly better off following the strategy prescribed to him. We denote by t the number of the current iteration, and by s_i the size of the share assigned to player i. Consider the following cases:

Case 1. Player i's share has ended ($t = s_i + 1$)

In this case, player i knows he has the shorter share. By deviating, he only retrieves the secret if the current stage happens to be the last in the iteration, or if he successfully guesses the secret. With probability $1 - \beta$, the current stage is not the last, and since i's share and the current transcript do not convey any information about the secret, i's best guess is b. Denote by α' the probability that i guesses the secret correctly if he deviates in this stage. α' satisfies:

$$\alpha' \leq \beta + (1-\beta) \cdot \mathcal{D}(b) = \beta(1 - \mathcal{D}(b)) + \mathcal{D}(b)$$

As shown before, it suffices to demand:

$$
\begin{aligned}
\alpha' &< c_i \\
\beta(1-\mathcal{D}(b)) + \mathcal{D}(b) &< c_i \\
\beta &< \frac{c_i - \mathcal{D}(b)}{1-\mathcal{D}(b)} \\
\beta &< \beta_0
\end{aligned}
$$

Case 2. Player i's share has not ended ($1 \le t \le s_i$)

Denote by $\mathsf{definitive\_prob}$ the probability that the current iteration is definitive, given that player i reached the information set I containing the history h (in other words, $\mathsf{definitive\_prob}$ is the probability that the current iteration is definitive when given player i's share and the transcript so far). We first show that $\mathsf{definitive\_prob}$ is rather small. Intuitively, if s_i is close t then i is fairly convinced that he has the short share, and thus this iteration is probably not definitive. If s_i is a lot larger than t, player i believes he has a long share, but then any future iteration might be the definitive one.

Claim E.3. $\mathsf{definitive\_prob} \le \frac{z \cdot n \cdot \beta}{(1-\beta)}$.

Proof The only parameters viewed by i that are relevant when determining whether the current iteration is definitive, are: the number of the current iteration t, i's share size s_i, and the current unmasked possible secret y_t (learned by player i after iteration $t-1$). All the other values viewed by player i are independent of the secret and its revelation time. Assume that $s_i = k$ ($k > t$), $y_t = a$, and that player i_0 is the one with the short share.

$$
\begin{aligned}
\mathsf{definitive\_prob} &= \Pr\left[\ell = t \mid \ell \ge t \wedge y_t = a \wedge s_i = k\right] \\
&= \frac{\Pr\left[\ell = t \wedge y_t = a \wedge s_i = k\right]}{\Pr\left[\ell \ge t \wedge y_t = a \wedge s_i = k\right]} \\
&= \frac{\Pr\left[i \ne i_0 \wedge \ell = t \wedge y_t = a \wedge s_i = k\right]}{\Pr\left[i = i_0 \wedge \ell \ge t \wedge y_t = a \wedge s_i = k\right] + \Pr\left[i \ne i_0 \wedge \ell \ge t \wedge y_t = a \wedge s_i = k\right]}
\end{aligned}
$$

We calculate the probabilities appearing in the last term. Recall that if t is the definitive iteration ($t = \ell$) then $y_t = y$ where y is the real secret, otherwise $y_t = r_t$ for a randomly chosen $r_t \in Y$.

The term in the numerator:

$$
\begin{aligned}
& \Pr\left[i \ne i_0 \wedge \ell = t \wedge y_t = a \wedge s_i = k\right] \\
=\ & \Pr[i \ne i_0] \cdot \Pr[\ell = t] \cdot \Pr[y = a] \cdot \Pr[d = k - t + 1] \\
=\ & \frac{n-1}{n} \cdot \beta(1-\beta)^{t-1} \cdot \mathcal{D}(a) \cdot \beta(1-\beta)^{k-t} \\
=\ & \frac{n-1}{n} \cdot \mathcal{D}(a) \cdot \beta^2 (1-\beta)^{k-1}
\end{aligned}
$$

The first term in the denominator:

$$
\begin{aligned}
& \Pr\left[i = i_0 \wedge \ell \ge t \wedge y_t = a \wedge s_i = k\right] \\
=\ & \Pr[i = i_0] \cdot \Pr[r_t = a] \cdot \Pr[\ell = k + 1] \\
=\ & \frac{1}{n} \cdot \frac{1}{z} \cdot \beta(1-\beta)^k
\end{aligned}
$$

The second term in the denominator:

$$\begin{aligned}
& \Pr\left[i \neq i_0 \wedge \ell \geq t \wedge y_t = a \wedge s_i = k\right] \\
= \; & \Pr[i \neq i_0] \cdot \Pr[r_t = a] \cdot \sum_{j=t+1}^{k} \Pr[\ell = j] \cdot \Pr[d = k - j + 1] \text{ (t is not definitive)} + \\
& \Pr[i \neq i_0] \cdot \Pr[y = a] \cdot \Pr[\ell = t] \cdot \Pr[d = k - t + 1] \text{ (t is definitive)} \\
= \; & \frac{n-1}{n} \cdot \frac{1}{z} \cdot \sum_{j=t+1}^{k} \beta(1-\beta)^{j-1} \cdot \beta(1-\beta)^{k-j} + \\
& \frac{n-1}{n} \cdot \mathcal{D}(a) \cdot \beta(1-\beta)^{t-1} \cdot \beta(1-\beta)^{k-t} \\
= \; & \frac{n-1}{n} \cdot \beta^2(1-\beta)^{k-1} \left(\frac{1}{z} \cdot (k-t) + \mathcal{D}(a)\right)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathsf{definitive\_prob} & = \frac{\frac{n-1}{n} \cdot \mathcal{D}(a) \cdot \beta^2(1-\beta)^{k-1}}{\frac{1}{n} \cdot \frac{1}{z} \cdot \beta(1-\beta)^k + \frac{n-1}{n} \cdot \beta^2(1-\beta)^{k-1}\left(\frac{1}{z} \cdot (k-t) + \mathcal{D}(a)\right)} \\
& = \frac{(n-1) \cdot \mathcal{D}(a) \cdot \beta}{\frac{1}{z} \cdot (1-\beta) + (n-1) \cdot \beta\left(\frac{1}{z} \cdot (k-t) + \mathcal{D}(a)\right)} \\
& \leq \frac{z \cdot n \cdot \beta}{(1-\beta)}
\end{aligned}$$

■

Next, we show that player i's ability to guess the correct secret was not significantly improved from the beginning of the game. Denote by $\mathcal{D}'$ the distribution over the secrets induced by the information set I of player i containing the history h (in other word, $\mathcal{D}'$ is the distribution over secrets when given player i's share and the transcript so far). Note that $\mathcal{D}'$ may be different from the original $\mathcal{D}$: if the current unmasked possible secret is y_t, the probability that the real secret is y_t increases. For the time being, player i's best guess is the element $b' \in Y$ with the highest probability according to $\mathcal{D}'$. Therefore, the higher $\mathcal{D}'(b')$ is, the better i's ability to guess correctly.

Claim E.4. $\mathcal{D}'(b') \leq \mathsf{definitive\_prob} + \mathcal{D}(b)$.

Proof It is easy to see that $\mathcal{D}'(b')$ is maximal when $y_t = b$, in such a case $b' = b$. Thus:

$$\begin{aligned}
\mathcal{D}'(b') & \leq \Pr\left[y = b \mid \ell \geq t \wedge y_t = b \wedge s_i = k\right] \\
& = \Pr\left[y = b \wedge \ell = t \mid \ell \geq t \wedge y_t = b \wedge s_i = k\right] + \\
& \quad\; \Pr\left[y = b \wedge \ell \neq t \mid \ell \geq t \wedge y_t = b \wedge s_i = k\right] \\
& \leq \mathsf{definitive\_prob} + \mathcal{D}(b)
\end{aligned}$$

■

Finally, if player i deviates in the current iteration he is able to retrieve the secret only when one of the following occurs: the current iteration is definitive; he was not caught cheating; or he was able to guess the correct value. As before, we require:

$$\begin{aligned}
\mathsf{definitive\_prob} + \beta + \mathcal{D}'(b') &< c_i \\
2 \cdot \mathsf{definitive\_prob} + \beta + \mathcal{D}(b) &< c_i \\
\frac{2z \cdot n \cdot \beta}{(1-\beta)} + \beta &< c_i - \mathcal{D}(b) \\
2z \cdot n \cdot \beta + (1-\beta)\beta &< (1-\beta)(c_i - \mathcal{D}(b)) \\
\beta(2z \cdot n + 1 + (c_i - \mathcal{D}(b))) &< c_i - \mathcal{D}(b) \\
\beta &< \frac{c_i - \mathcal{D}(b)}{(c_i - 2\mathcal{D}(b)) + 2z \cdot n + 1} \\
\beta &< \beta_0
\end{aligned}$$

■

F An ε-Rational Secret Sharing Scheme for the NSBC Model

Formal description of the scheme. A formal description of the dealer's algorithm described in Section 6 can be found in Figure 6. The changes made in previous the algorithm are emphasized.

Some additional notes. The following are some additional notes regarding the suggested scheme:

Note F.1. In the new protocol we cannot prevent the short player from deviating when the definitive iteration is reached by dividing every iteration into separate stages, therefore the iterations are no longer divided. Instead, the protocol makes sure that the short player's deviation will be detected with a high probability.

Note F.2. 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 an ε-everlasting equilibium, and is susceptible to backward induction.

Dealer(y, β, ε)

Denote by Π_n ***the set of all permutation on*** n ***elements. Let*** $\mathbb{F}' = GF(p)$ ***for*** $p \geq |Y \times \Pi_n|$ ***prime, and identify each pair of secret and permutation in*** $Y \times \Pi_n$ ***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 possible secrets.
  - Select at random a list of size L of possible secrets (elements of Y), such that its ℓ^{th} element is y**.**
  - ***Select at random a list of size*** L ***of permutations over*** n ***elements.***
- **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*** $\mathcal{G}(\beta)$***.***~~ ***(see Note F.1)***
  - **Mask**: An m-out-of-n Shamir share of a randomly chosen element of $\mathbb{F}'$ used to mask the *next* possible secret ***and permutation***.
  - **Masked secret *and permutation***: An element of $\mathbb{F}'$ obtained by summing, over $\mathbb{F}'$, the pair of corresponding elements in the secrets ***and permutations*** lists and the mask shared between the players in the *previous cells*.
  - **Indicator** (exists only for $n > m$): An m-out-of-n Shamir share of a boolean value indicating whether this iteration is definitive.
  - **Authentication information**: A "tag" allowing the player to prove the authenticity the previous elements in this cell, and a "hash function" allowing him to check the authenticity of elements in the corresponding cells of the other vectors with probability at least $\mathbf{1 - \varepsilon'}$ ***for*** $\boldsymbol{\varepsilon' = \min\{\beta, \frac{\varepsilon}{U_{max}}\}}$ ***where*** U_{max} ***is an upper bound on the payoffs that the players may receive*** (can be achieved with tag and hash of size $\log \frac{1}{\varepsilon'}$).

  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**: **Give the short share to the player chosen to broadcast last according to the*** ℓ^{th} ***permutation. Choose a random assignment of the remaining shares to the other players.***

Figure 6: The dealer's shares assignment algorithm for the NSBC model
The changes made in previous algorithm are emphasized in blue