Equilibrium in an Asynchronous Model

Yoram Moses*

Rann Smorodinsky†

September 8, 2003

Abstract

We embed a model of normal form games, where players act simultaneously, in a standard model of an asynchronous distributed system, where simultaneous play is impossible due to the lack of a global clock. We show that despite the lack of simultaneous play, the concept of Nash equilibrium can be salvaged.

1 INTRODUCTION

The standard model of non-cooperative games with complete information is that of *extensive form games*. Such modelling enables us to analyze the interaction among players who move simultaneously or at different times (turns) and who have varying degrees of information on past actions. Among the set of such games, the subset assuming all players have a single turn and move simultaneously is most widely explored.

*Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel. <moses@ee.technion.ac.il>

†Support from the fund for the promotion of research at the Technion and the William Davidson Research fund is gratefully acknowledged. Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel. <rann@ie.technion.ac.il>

Such simultaneous move interactions are often modelled in matrix form and referred to as games in *normal form*.

In the analysis of games in normal form, the notion of Nash equilibrium is used to predict and explain behavior. Underlying the use of Nash equilibrium (and in particular that of a mixed-strategy equilibrium) is the assumption that the opponents' choice of action is not available to a player when he takes his action. In reality, this can be achieved if players play simultaneously, or, alternatively, if another trustworthy party is involved in the game. This new player discretely collects all players' actions and reveals them (or just the final outcome of the chosen actions) truthfully only after all players have submitted their actions. In particular, it is assumed that the new player has no incentives of his own.

Whereas the latter explanation has an obvious weakness (ie., the need to assume a particular type of player $N+1$ when modelling an N-player game), the former explanation, on the other hand, has traditionally been satisfactory for modelling economic markets, political conflicts, etc. However, when one considers the computer-based forms of interaction available today, the need to enforce simultaneous action becomes a major limitation.

Example 1 *Consider two players engaged in a game of matching pennies, via the internet (for simplicity, think of the game played via e-mail). If they have no third party to rely on, then suddenly the* $(0.5, 0.5)$ *strategy is not an equilibrium strategy. In fact, assuming that players cannot coordinate on the same split second for announcing their respective chosen actions, the optimal behavior of the player that matches, assuming his opponent does play* $(0.5, 0.5)$*, is to wait and see his opponent's chosen action and immediately reply with an action that matches it.*

In fact, this paper deals with a problem raised when one considers the actual implementation of a game (in normal form) over a network of computers. Thus, our paper joins the recent strand of literature on 'internet games'. This literature has focused on analyzing games played by economic agents, interacting via a "web" of computational

devices (i.e., a distributed system). The introduction of a distributed system adds various 'engineering' difficulties to the analysis of games. Some examples for this are the difficulties due to the communication structure (e.g., Monderer and Tennenholtz [18]), computational complexity (e.g., Nissan [21]), issues of communication complexity (e.g., Shoham and Tennenholtz [27] and Holzman *et al.* [14]), cost of communication (e.g., Blumrosen and Nissan [6]), etc. This paper focuses on the inherent difficulty in a practical setting to enforce truly simultaneous actions by the different players. In fact, in a realistic model of a distributed system, it is impossible to perfectly synchronize clocks, and so players cannot fully coordinate on some pre-specified time (even if they all wish to do so).

For game theorists, the most fundamental result in the theory of normal form games is the existence of a Nash equilibrium. The standard proof of this result depends, however, in a strong way on the assumption that players move simultaneously. This suggests that, in the absence of a trusted incentive-free party, the existence of equilibria may be jeopardized once we take into account the lack of simultaneity which is inherent in distributed systems.

In this paper we show that not all is lost. The main contributions of this paper can be summarized as follows.

- Within the context of a distributed computer system, we define a setting that allows an analysis of behavior in terms of selfish interest and rational decision making.
- In this setting, we define solution concepts that are the asynchronous versions of various types of equilibria in normal form games, and in particular we define the asynchronous equivalent of a Nash equilibrium.
- Finally, we describe ways to embed the strategic structure of a game in normal form in a distributed system setting, and show that strategies in the normal form game can be "translated" into protocols in the distributed system. Different embeddings are shown to preserve different notions of equilibrium. Thus,

for example, strategies that are in Nash equilibrium in the game translate into protocols that are in (protocol-)Nash equilibrium in the distributed system. This shows both that the concepts of equilibrium in distributed systems are not vacuous, and that the traditional game theory literature promises to be applicable in a practical distributed system setting.

From the computer-scientific perspective, this paper focuses on how a player can commit to his strategy without revealing it. While this question has received extensive treatment in the area of cryptography (e.g., Goldreich *et al.* [9]), the economic dimension changes the problem in some ways. Specifically, we restrict attention to deviations from the protocol that offer the deviating players a positive expected utility. Such "rationality" criteria impose weaker constraints on the protocol than standard criteria used in cryptography and in the literature on fault-tolerance in distributed computing. Consequently, the protocols complying with them can be simpler.

The normal form model is not applicable to the context of asynchronous distributed protocols. Thus, the framework we choose for describing asynchronous games is the one used by researchers dealing with distributed systems. It is worth noting that the model of extensive form games provides a setting that is somewhat closer in spirit to distributed protocols. However, our modelling choice has two notable properties:

- Cultural - The natural and standard way to model behavior in a network is by way of protocols. By relating games to protocols, we are striving to make a small contribution to the task of bridging the cultural gap between the scientific disciplines that treat games and those that design distributed protocols.

- Randomness - In extensive form games, each random choice is modelled by way of a random variable. Such a random variable is motivated by an objective prior (e.g., the roll of a die) or by subjective beliefs. However, in a computer network, nature will typically play a compound and multi-dimensional role. It can govern standard probabilistic events such as a lottery, but it will also affect when players can move, how long it will take for messages to reach their destinations, etc. We

will typically be able to provide the protocol with probabilistic priors regarding some of the events at hand, such as lotteries, while priors for other aspects of nature's behavior may not be available. Our definitions and the modelling of distributed systems that we use will facilitate such a view. The analysis we provide does not assume anything on this randomness. Indeed, most protocols are expected to be used in many different instances and settings, without the need to obtain detailed statistics about the properties of nature. The definitions provided in this paper and the notions of equilibrium considered have the property that they depend only on the player's randomization, and are robust under a wide variety of choices of nature's properties. Thus, our protocols are often guaranteed to be in equilibrium regardless of how nature schedules the timing of message delivery and/or the rate at which players operate.

1.1 An Anecdotal Story of an Asynchronous Setting

To motivate the study of asynchronous settings we refer to a story (possibly a myth) on the accumulation of wealth by the Rothschild family, as told by SEC Staff, Paul F. Roye (Director, Division of Investment Management) in a speech carried out in the 2000 Mutual Funds and Investment Management Conference Palm Desert, Calif., [23]:[1]

"In following such distinguished speakers as Marianne Smythe and Matt Fink, I pondered long and hard what I could say that would be as profound and insightful as their remarks. After some thought, I determined that I would focus on what we might learn from the use of "pigeons," a much maligned bird, which I have often heard referred to as "rats with wings." I don't even think they have pigeons in Palm Desert they can't afford the nests. But seriously, Nathan Rothschild, one of the greatest financiers in history, discovered that he could use carrier pigeons to obtain a higher quality of information, faster than anyone else. These carrier pigeons, which were like an early intranet, or a pre-electronic e-mail system, kept Rothschild hours, days, even

[1]The story is also told in Fredric Morton's, *The Rothschilds - A Family portrait*, Atheneum New York ,1962.

weeks ahead of what his competitors were doing in the early banking days in Europe. As a result, he utilized that information to make better financial decisions.

You history buffs know that the battle of Waterloo was fought in June of 1815. In February of 1815, Napoleon escaped from Elba and assembled what was viewed as an invincible army. The Allies, having fought Napoleon's armies for fifteen years, were nearly bankrupt. The Allies, however, conscripted another rag-tag army and sent it to Belgium under Wellington's command to meet Napoleon's large and seasoned on-coming forces. But no one including those in the financial markets believed the Allied Army would defeat Napoleon.

Napoleon's victory seemed all too likely and the price of British government bonds reflected that prospect. Now we all know what the outcome of the Battle of Waterloo was but so did Nathan Rothschild. As the story goes, flocks of carrier pigeons kept him apprised of the battle, allowing him to buy British government bonds at rock bottom prices, thereby enabling him to create another fortune when news of the victory arrived by normal means to Britain's shores. Rothschild realized that using the most advanced means available to quickly secure information was critical to success in the securities markets. Then, as now, innovation was key. But even the Rothschilds were overtaken by it. Their control over the flow of information through a network of couriers and carrier pigeons was so effective for a time, that Europe's heads of state relied on it. But after the telegraph rendered this network obsolete, James Rothschild, brother of Nathan, complained that now "anyone can get the news." It is clear that the mutual fund industry must innovate, embrace technology and use it to remain competitive."

The story above is an excellent example of the possible implications of asymmetric information that is induced by asynchronous systems. In this example it was commonly known that Nathan Rothschild had the information first. However in today's electronic networks there is non commonly known agent which has an advantage. It depends on uncontrolled parameters of the information network. Our model accounts for this consideration.

The next section presents the model of communication and asynchronous games. Section 3 provides the main results and section 4 discusses the intuition behind the

results. Section 5 compares the model and results to related literature and, finally, section 6 discusses directions for future research.

2 THE MODEL

The model we describe has three components. First, the game-theoretic component that consists of a game in normal form. This is given by a set of strategies available to each player and the payoff function which maps tuples of players' strategies into payments. Second, we have a communication network over which the players interact. This is an asynchronous distributed system that contains a shared resource in the form of a public electronic bulletin board. Corresponding to a player in the game, we have a process that represents this player in the distributed system, and the analogue of a strategy for the player is a *protocol* for the corresponding process. Finally, our model describes a way of mapping the game into the distributed system, so that behavior that corresponds to a possible choice of strategies in the original game yields similar payoffs to the participants in the distributed system.

2.1 The Game

Let $N = \{1, 2, \ldots, n\}$ be the set of players. For each player $i \in N$, let S_i denote the (finite) set of strategies available to i. For ease of exposition, we assume without loss of generality that $S_i = S = \{s_1, s_2, \ldots, s_K\}$ for all $i \in N$. Let S^n be the set of all strategy tuples. For every $i \in N$, let $U_i : S^n \rightarrow R$ denote player i's payoff function. A *game* is defined by a pair $G = (S^n, \{U_i\}_{i=1}^n)$, where S^n is as before, and $\{U_i\}_{i=1}^n$ is a vector of payoff functions, one for each player. A game is said to be *individually rational* (IR) if each of the U_i's is strictly positive (i.e., $U_i(s) > 0$ for all $s \in S^n$ and all i). In an IR game, every player gains something by participating in the game. Thus, the players all have an incentive to play, if we assume that non-participation yields a payment of zero. In this paper we will focus on IR games. One of the consequences will be that in

a distributed computation corresponding to a game, no participant will be motivated to opt out by stopping to play his role.

A mixed strategy for player i is a probability choice, $\sigma_i \in \Delta(S_i)$, over the set S. A *Nash equilibrium* is a vector of strategies $\{\sigma_i\}_{i=1}^n$, such that

$$\sigma_i \in \arg \max_{\tau \in \Delta(S_i)} E_{\tau,\sigma_{-i}}(U_i(s_i, s_{-i})).$$

In other words, if each player chooses a strategy according to the distribution of σ_i, then no player has an incentive to deviate.

2.2 Communication

The notion of Nash equilibrium is valid in a world where all players announce their chosen strategies prior to learning anything about other players' strategies. One possible implementation of such a constraint is by having players announce their strategies simultaneously. However, it is more realistic to assume that players do not have a global clock and therefore cannot choose or announce strategies simultaneously. The problem becomes more acute when the players interact over a computer network, in which case it is reasonable to assume that all actions are asynchronous. This is the setting that we are interested in modelling now.

We assume that the players have two ways to communicate with each other: First, there are private point-to-point communication channels over which pairs of players can communicate. Every message sent via such a channel is guaranteed to be delivered intact eventually. Moreover, the messages that a player sends over a given channel are guaranteed to be delivered in the order in which they are sent. There is no *a priori* bound on the time the message spends in transit, however (this is called *asynchronous* communication.) Second, there is a *bulletin board* (BB) to which players may post a message.[2] Posting a message m on the bulletin board is done by sending a $\mathsf{post}(m)$ message to the board. We assume that the bulletin board consists of the list of messages

[2]E.g., chat or message board, as are currently popular on the internet.

that were posted to the board, in the order they arrived at the board; no messages are ever erased or deleted from the bulletin board. To read (perhaps an update on) the messages posted on the bulletin board, a process sends a read message to the board. Once this message arrives, the board immediately responds by sending the process a message describing the current contents of the board. Just like communication between players, communication to and from the BB is assumed to be reliable but asynchronous.

2.3 Protocols

The notion of a *protocol* in a distributed system plays a role similar to that of a *strategy* in a game. Since we think of an interaction in a distributed system as consisting of an execution that proceeds over a number of steps, this is closer to a strategy in a game in extensive form. Intuitively, a protocol specifies the actions that a process (or player; we use the terms interchangeably) should take, at every stage of the execution.

In order to formalize this, we ascribe to every process a *local state* at any instant of time. In addition, each process i has a set A_i of potential *actions* it can perform. In our case, an action $a \in A_i$ can include one or more read, post or send actions, as well as local steps. It is also convenient to assume that a process can perform a halt action, after which it performs no action. A *deterministic protocol* for process i is a function $D_i : L_i \to A_i$, from the set L_i of local states of i to A_i.

A *joint deterministic protocol* is a tuple $D = (D_1, \ldots, D_n)$ consisting of a protocol for each process. We denote by D the set of (individual) deterministic protocols for the processes.

Often protocols are based on random choices (e.g., an execution of a mixed strategy). Intuitively, this means a process should have access to a source of randomness. We can think of a process using randomization as performing a preliminary action of choosing a probability space, $(\Omega_i, \mathcal{F}_i, Q_i)$, where $\mathcal{F}_i$ is a σ-field over a state space Ω_i and Q_i is a probability measure. As a result, a certain ω_i is realized that the player can access.

Formally, we define a (randomized) *protocol* for process i, to be a pair $\mathcal{P}_i = (P_i, (\Omega_i, \mathcal{F}_i, Q_i))$, consisting of a probability space $(\Omega_i, \mathcal{F}_i, Q_i)$, and a function, P_i, from all states in Ω_i into the set of deterministic protocols for i. Let $P_i(\omega_i)$ denote the deterministic protocol at ω_i and let $P(\omega)$ be the joint deterministic protocol $(P_1(\omega_1), \ldots, P_n(\omega_n))$, where $\omega = (\omega_1, \ldots, \omega_n)$.

Without loss of generality, we assume that processes have perfect recall. We capture this formally by assuming that the local state of a process is a list consisting of all actions it performed and all messages received (content, source, and type). The items in the list appear in the order in which they took place. Our assumptions imply, in particular, that the processes start out with an empty list as their initial local state.

The *local history* h_i of a process i is defined to be the sequence of local states it goes through. This records all changes that the process is able to observe. Similarly, the local history h_{BB} of the BB is the sequence of states the bulletin board goes through over time. A *run* captures the history of a computation. It is defined to be an $(n+1)$-tuple $r = (h_{BB}, h_1, \ldots, h_n)$ consisting of the local histories of the BB and of all n processes.

2.4 Nature's Role

While deterministic protocols resemble the actions (or pure strategies) available to players, protocols resemble strategies in a game, and runs resemble the history of a game, there is a subtle point worth noting. We are implicitly assuming the existence of a "hidden" player participating in the execution of any deterministic protocol in a distributed system. We may call it *nature* (or alternatively others refer to it as the *environment*, or the *scheduler*). Its role is to determine various aspects of the run that the deterministic protocols do not control. In our case, these are the lengths of time that each message spends in transit, and the times at which each of the processes "moves". The deterministic protocol of a process is applied to its local state only when the process is scheduled to move. We will assume that nature schedules every process

to move an infinite number of times in any given run. (Of course, a process that has halted will not perform an action when it is scheduled.)

Formally, an action of nature is composed of an infinite sequence of the form $\{(t_k, \phi(t_k), \{\tau_m(t_k)\}_{m=0}^n)\}_{k=1}^\infty$, where $\{t_k\}_{k=1}^\infty$ is an increasing sequence of positive numbers, denoting the times (according to nature's own internal clock) at which players' moves take place. $\phi(t_k) \in N$ is the player assigned to move at time t_k. We require ϕ to be such that every player is assigned a move infinitely often. $\{\tau_m(t_k)\}_{m=0}^n$ is a finite sequence of positive numbers denoting the following: For $m = \phi(t_k)$, $\tau_m(t_k)$ denotes the processing time needed for player $\phi(t_k)$ to perform its move at time t_k (it completes the move at time $t_k + \tau_m(t_k)$). For a player $m \neq \phi(t_k)$, $\tau_m(t_k)$ denotes the time that it will take a message sent by $\phi(t_k)$ to m, in case one is sent in its move at t_k, to be delivered (again, the message is delivered to m at time $t_k + \tau_m(t_k)$). Finally, $\tau_0(t_k)$ denotes the time that it will take a message sent by $\phi(t_k)$ to the BB, in case one is sent in its move at t_k, to be delivered (the message is delivered to the BB at time $t_k + \tau_0(t_k)$). We shall denote by $\mathcal{A}$ the set of all possible actions of nature.

Given the powers of nature to affect the course of events, a given set of deterministic protocols $D = (D_1, \ldots, D_n)$ for the n players does not uniquely determine a run that will result when the players follow their respective protocols. Rather, it determines a set of runs, which we will denote by $R(D)$. However, once nature's action, $\gamma \in \mathcal{A}$, is determined, a unique run, denoted $r(D, \gamma)$, is defined. This fact will have an impact on how we relate the strategies in our game to protocols, which is the subject of the next section.

2.5 An Asynchronous Model of a Game

Formally, a *clearing mechanism*, $V = (V_1, V_2, \ldots, V_n)$, is a mapping from the set of possible runs into individual payoffs. It ascribes to each player i its payoff $V_i(r)$ in the run r. For the purposes of this paper, we assume that the outcome of the clearing mechanism depends only on the local history of the BB in the run.

A (joint) protocol $(\mathcal{P}_i)_{i=1}^n$ is called a *Nash equilibrium* with respect to a clearing mechanism V if for every pair of actions of nature, $\gamma^1, \gamma^2 \in \mathcal{A}$, every process i and protocol $\hat{\mathcal{P}}_i = (\hat{P}_i, (\hat{\Omega}_i, \hat{\mathcal{F}}_i, \hat{Q}_i))$ for i,

$$E_{(Q_1,\ldots,Q_n)}[V_i(r(P(\omega), \gamma^1))] \geq E_{(\hat{Q}_i, Q_{-i})}[V_i(r((\hat{P}_i(\hat{\omega}_i), P_{-i}(\omega_{-i})), \gamma^2))].$$

Thus, in a Nash equilibrium, *regardless of nature's behavior in the run*, no process can expect to gain by deviating from its protocol if all the others behave according to the joint protocol.[3]

2.6 Embedding Games in an Asynchronous Model

Let $G = (S, U)$ be an arbitrary game in normal form. Our goal is to describe a mapping from mixed strategies in G to protocols that the players could use in order to, intuitively, simulate a play of the game on an asynchronous system. Rather than working at the level of mixed strategies, we will typically map each of the strategies in S (the pure strategies) to a protocol. To obtain behavior akin to that of a mixed strategy, we will have the player mix over the protocols corresponding to the pure strategies. More formally, if $\pi(s_i) = (P_{s_i}, (\Omega_{s_i}, \mathcal{F}_{s_i}, Q_{s_i}))$ is the (possibly randomized) protocol corresponding to pure strategy s_i, then the protocol $(P_{\sigma_i}, (\Omega_{\sigma_i}, \mathcal{F}_{\sigma_i}, Q_{\sigma_i}))$ corresponding to a mixed strategy σ_i is described as follows. $\Omega_{\sigma_i} = \{(s_i, \omega_{s_i}) : s_i \in S_i \text{ and } \omega_{s_i} \in \Omega_{s_i}\}$, $Q_{\sigma_i}(s_i, \omega_{s_i}) = \sigma_i(s_i) Q_{s_i}(\omega_{s_i})$ and $P_{\sigma_i}(s_i, \omega_{s_i}) = P_{s_i}(\omega_{s_i})$. One can think of this as a protocol by which player i chooses a strategy s_i according to σ_i as its first step, and then follows $\pi(s_i)$. The result is clearly a randomized protocol. In this fashion, a mapping π associating a protocol to each pure strategy in S induces a mapping from every mixed strategy σ_i to a protocol $P_{\pi(\sigma_i)}$.

The following lemma captures the idea that the induced protocol behaves as intended:

[3] A natural way to weaken this definition is to assume a distribution over the way runs are determined and take an expectation in the inequality. Note that the suggested definition circumvents the need to model nature's random behavior explicitly.

Lemma 1 *Let $G = (S, U)$ be a game, and let π be a mapping from S to protocols. If*

$$U(s) = E_{Q_{\pi(s)}}[V_i(r(P_{\pi(s)}(\cdot), \gamma))]$$

for every action of nature γ and every pure strategy tuple $s = (s_1, \ldots, s_n) \in S^n$, then

$$E_\sigma(U(\cdot)) = E_{Q_{\pi(\sigma)}}[V_i(r(P_{\pi(\sigma)}(\cdot), \gamma))]$$

for every action γ and mixed strategy tuple $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Delta(S^n)$.

Proof Fix γ and suppose $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Delta(S^n)$. In the initial step of a run of $\pi(\sigma)$, each player i chooses a strategy independently according to σ_i. The tuple of strategies is obtained according to probability distribution σ. For every tuple s of strategies thus chosen, we have by assumption that $E_{Q_{\pi(s)}}[V_i(r(P_{\pi(s)}(\cdot), \gamma))] = U(s)$. It follows that

$$E_{Q_{\pi(\sigma)}}[V_i(r(P_{\pi(\sigma)}(\cdot), \gamma))] = E_{(s \in \sigma)}(E_{Q_{\pi(s)}}[V_i(r(P_{\pi(s)}(\cdot), \gamma))]) = E_{(s \in \sigma)}(U(s)) = E_\sigma(U(\cdot)),$$

as desired. □

Definition 1 *Let V be a clearing mechanism, and let $\pi = (\pi_1, \ldots, \pi_n)$, where each π_i is a mapping from S to protocols. The pair (π, V), is a* **proper representation** *of the game $G = (S, U)$ if*

$$E_{Q_{\pi(s)}}[V(P_{\pi(s)}(\cdot), \gamma)] = U(s)$$

holds for all $s \in S^n$ and all actions $\gamma \in \mathcal{A}$.

In other words, a proper representation is a representation that preserves the payoff structure of any strategy tuple, regardless of what action nature ends up performing.

3 MAIN RESULTS

The main issue we cope with in this article, beyond a definition of an asynchronous game, is the possibility (or impossibility) of having proper representations for normal

form games in an asynchronous setting such that the major game theoretic concepts are preserved. In this chapter we provide some positive results. All the results are based on existing ideas from the theory of cryptography.

Our first result shows that for games of three or more players, a proper representation exists for which Nash equilibria are preserved. The proof of this result is based on a simple application of "secret-sharing." Using the same concept, we can extend the result to preserve other notions of equilibria, namely N-equilibrium (where deviations of coalitions of size N or less are not incentive compatible) and Correlated equilibria.

3.1 Nash Equilibrium

The following theorem shows that there is a general scheme to embed a standard game model in its strategic form into an asynchronous extension of the game, in a way that preserves Nash equilibria.

Theorem 1 *Let G be an IR game of $n \geq 3$ players. There is a proper representation $G_{asynch} = (\pi, V)$ of G such that for every Nash equilibrium $\{\sigma_i\}_{i=1}^n$ of G, the corresponding protocol $\{\pi(\sigma_i)\}_{i=1}^n$ is a Nash equilibrium of (π, V).*

Proof: We will start by defining a mapping π for pure strategies, and show that $U(s) = E_{Q_{\pi(s)}}[V_i(r(P_{\pi(s)}(\cdot), \gamma))]$ for every action of nature γ and every pure strategy tuple $s = (s_1, \ldots, s_n) \in S^n$.

Step 1 - Commitment.

This step is based on a simple but elegant scheme for secret sharing. Recall that s_i is an integer between 1 and $K = |S|$.

(i) Send α_i to player $i+1$ and $\beta_i = \alpha_i + s_i \bmod(K+1)$ to player $i+2$. Recall that $n \geq 3$ and so i, $i+1$ and $i+2$ are 3 different players.[4] Note that the posterior distribution for players $i+1$ and $i+2$ regarding player i's chosen strategy, given their information at any point in time during the first step, equals the prior.

[4]Obviously, the computation of $i+1$ and $i+2$ is done modulo n, the number of players.

(ii) Wait for a message from $i-1$ containing some number α'_{i-1}, and for a message from $i-2$ containing a number β'_{i-2}. Then send a "ready" message to the bulletin board.

Step 2 - Announcements.

Wait to see a "ready" message from all n players posted on the BB. Then send a message stating $(s_i, \alpha'_{i-1}, \beta'_{i-2})$ to the bulletin board.

Step 3 - Final step.

Wait until a message of the form $(s_j, \alpha'_{j-1}, \beta'_{j-2})$ is on the BB, for all j, then halt.

This completes the description of the protocol $\pi_i(s_i)$.

We now define the clearing functions V_i. Let R be a particular run of this protocol. We call the depiction of a player j by the BB *consistent* if it has only posted one message on the BB of the form $(s_j, \alpha'_{j-1}, \beta'_{j-2})$, and if the strategy s_j it reports is compatible with the values of α'_j and β'_j that are reported on its behalf.

- If the depictions of all players by the BB are consistent, then let $V_i(r) = U_i(s)$, where $s = (s_1, \ldots, s_n)$ is the sequence of strategies reported by players $1, \ldots, n$ respectively.
- Otherwise set $V_i(r) = 0$ for $i = 1, \ldots, n$.

Notice that the above clearing mechanism depends, as required, only on the history of the BB. [5]

Obviously, if $\sigma = \{\sigma_i\}_{i=1}^n$ is a Nash equilibrium of G and all players follow the suggested protocol, then the distribution of the outcome in G_{asynch} follows σ. It is

[5]Notice that this clearing mechanism guarantees that all players have access to the outcome of the run, since they all receive reports describing the history of the BB.

now sufficient to show that no player has an incentive to deviate from the suggested protocol, assuming all other players follow that protocol.

Assume player i has sent messages α_i and β_i to players $i+1$ and $i+2$ respectively. It is obvious that from now on, it is optimal for him to follow the protocol and eventually submit to the BB, in the last step, the message $(\beta_i - \alpha_i, \alpha'_{i-1}, \beta'_{i-2})$. By doing so i expects to receive $E_{(\beta_i - \alpha_i, \sigma_{-i})}$, the expected value in the strategic form game G, where i plays the pure strategy $\beta_i - \alpha_i$ and all other players, denoted $-i$, play according to σ_{-i}. This is true because the secret sharing mechanism ensures that at the point in time i submits the messages β_i and α_i his distribution over the protocols of the strategy chosen by the other players is still σ_{-i}, even though he might have accumulated extra data, such as β_{i-1} or α_{i-2}. By deviating in a non-observed way i will not improve his situation, whereas by deviating in an observed way i can expect to receive zero.

We finish by noting that due to the properties of the Nash equilibrium, the above expectation is maximized (in the weak sense) in case s_i is chosen according to the distribution σ_i, α_i is chosen according to a uniform distribution and β_i is set to equal $\alpha_i + s_i \bmod (K+1)$. □

3.2 Correlated Equilibrium

Assume players' strategies are correlated. For example, consider a situation where a trusted party chooses a random outcome that has no impact on the payment structure (e.g., a sun-spot) and sends each of the players a signal about the chosen outcome. If players base their strategy choice on the signal, then their strategies become correlated. Aumann [1] captures a natural equilibrium for such a situation in his definition of a *correlated equilibrium*.

Barany [3] discusses correlated equilibrium, where the distribution has rational probabilities (we refer to this as *correlated equilibrium in rationals*). Barany provides a distributed protocol, for 4 players or more, where players can mimic the actions of the trusted party by communicating amongst themselves. Furthermore, any unilateral

deviation will either (a) go unnoticed by other players and will also not affect the marginal distribution of signals on the non-deviating players; or (b) will be noticed by all players.

Assume players engage as proposed by Barany. Once this distributed computation is done players proceed as follows: If a deviation was detected then all players halt. Otherwise players proceed according to the distributed protocol described in the proof of Theorem 1. This combined protocol provides a proof to the following:

Theorem 2 *Let G be an IR game of $n \geq 4$ players. There is a proper representation $G_{asynch} = (\pi, V)$ of G such that for every correlated equilibrium in rationals, $\{\sigma_i\}_{i=1}^n$ of G, the corresponding protocol $\{\pi(\sigma_i)\}_{i=1}^n$ is a correlated equilibrium in rationals of (π, V).*

3.3 $\bar{N}$-equilibria

The demand underlying the notion of a Nash equilibrium is that no single player will have an incentive to deviate. However, a Nash equilibrium does not rule out the possibility that a coalition consisting of more than one player may gain by (simultaneously) deviating from equilibrium play.

In a distributed system (one may think of a virtual community over the internet), it is natural to assume that coalitions may communicate without the knowledge of the other players, and consequently, may all deviate from their pre-designated protocol. However, it may be natural to assume that such coalitions are able to collude only if they are not too big. This induces the following definition. Let $\sigma = (\sigma_1, \ldots, \sigma_N)$ be a vector of strategies and let $\tau_C = (\tau_j)_{j \in C}$ be a collection of strategies for members in a coalition of players, $C \subset N$. We denote by (τ_C, σ_{-C}) the vector of strategies derived from σ by replacing the strategies of members of C by τ.

Definition 2 *A vector of strategies $\sigma = (\sigma_1, \ldots, \sigma_N)$ of a game G is an $\bar{N}$-equilibrium if for every coalition, C, of size $\bar{N}$ or less, and for every τ_C either $E_{(\tau_C, \sigma_{-C})}(U_j(s_C, s_{-C})) =$*

$E_\sigma(U_j(s_C, s_{-C}))$ $\forall j \in C$ *or there exists a member* j *of* C *such that*

$$E_{(\tau_C,\sigma_{-C})}(U_j(s_C, s_{-C})) < E_\sigma(U_j(s_C, s_{-C})).$$

We modify the notion of Nash equilibrium in protocols to $\bar{N}$-equilibrium in the following natural fashion:

Definition 3 *A (joint) protocol* $(\mathcal{P}_i)_{i=1}^n$ *is called an* $\bar{N}$-equilibrium *with respect to a clearing mechanism* V *if for every pair nature's actions,* $\gamma^1, \gamma^2 \in \mathcal{A}$*, every subset of processes* $C \subset N$ *and every set of protocols* $\hat{\mathcal{P}}_i = (\hat{P}_i, (\hat{\Omega}_i, \hat{\mathcal{F}}_i, \hat{Q}_i))$ *for* $i \in C$*, either*

$$E_{(Q_1,\ldots,Q_n)}[V_i(r(P(\omega), \gamma^1))] = E_{(\hat{Q}_C,Q_{-C})}[V_i(r((\hat{P}_C(\hat{\omega}_C), P_{-C}(\omega_{-C})), \gamma^2))]$$

or there exists a process $i \in C$ *for which*

$$E_{(Q_1,\ldots,Q_n)}[V_i(r(P(\omega), \gamma^1))] > E_{(\hat{Q}_C,Q_{-C})}[V_i(r((\hat{P}_C(\hat{\omega}_C), P_{-C}(\omega_{-C})), \gamma^2))].$$

The following is natural extension of Theorem 1:

Theorem 3 *Let* G *be an IR game of* $n \geq 3$ *players and let* $\bar{N} < n - 1$*. There is a proper representation* $G_{asynch}(\bar{N}) = (\pi, V)$ *of* G *such that for every* $\bar{N}$*-equilibrium* $\{\sigma_i\}_{i=1}^n$ *of* G*, the protocol* $\{\pi(\sigma_i)\}_{i=1}^n$ *is an* $\bar{N}$*-equilibrium of* G_{asynch}*. In fact, there exists a single representation satisfying the claim for all* $\bar{N} < n - 1$ *simultaneously.*

Proof: We adapt the techniques used to prove Theorem 1. To do so, we make use of a secret sharing scheme in which each player distributes his secret among a set of more than $\bar{N}$ players, in such a way that no coalition of size $\bar{N}$ or less can obtain any information about the secret. This can be implemented using the secret sharing scheme devised by Shamir ([26]). We now provide an overview of this scheme, which works for all cases of $\bar{N} < n - 1$. Recall that we assume the pure strategies are numbered $1, \ldots, K$. Since K is known in advance, we fix a prime number $q > \max\{K + 1, n\}$. Each process i (privately) chooses $n - 1$ natural numbers $a_1, \ldots, a_{n-1}$ in the range $[0, q - 1]$ independently and uniformly at random. This defines the polynomial

$$Q_i(x) = s_i + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}.$$

This is a polynomial whose free term is s_i—the index of i's chosen pure strategy. For every one of the processes $j \neq i$, process i evaluates $Q_i(x)$ at the point $x = j$ over the *finite field* F_q, thereby obtaining a "share" $(j, Q_i(j))$. Given all $n-1$ shares produced by Q_i, it is possible to reconstruct the polynomial completely, and recover s_i. However, any set of $\ell < n-1$ of the shares provides no information whatsoever about the identity of s_i (as proven in [26]). Formally speaking, at any point up to the end of the secret sharing phase, and every subset C of fewer than $n-1$ processes, the posterior on s_{-C}, given all of the information available to the players in C, is the same as the prior.

The rest of the protocol is completely analogous to that in the proof of Theorem 1. In the first step, every process constructs its polynomial as described above and sends its shares to the other processes. Once a process has sent its messages and has received a share from each of the remaining $n-1$ processes, it sends a "ready" message to the bulletin board. Once n "ready" posts appear on the board, the first step ends and the second step begins. In the second step, each process i posts a message of the form $(s_i, \{(j, i, Q_j(i))\}_{j \neq i})$ revealing both s_i and all $n-1$ shares it has received.

As before, the processes receive a payment of $V(s_1, \ldots, s_n)$ in case all of the information revealed is consistent (i.e., for every j the shares $(j, i, Q_j(i))$ posted on the BB extrapolate to a polynomial whose free term is equal to the posted s_j), and they all receive payments of 0 in case there is an inconsistency.

The secrecy guaranteed by Shamir's scheme makes the analysis of this protocol coincide with the one in the case of simple Nash equilibria. □

A *strong equilibrium* (see Aumann [2]) is an $\bar{N}$*-equilibrium* with $\bar{N} = N$, the number of players in the game. In other words, in a strong equilibrium no coalition has an incentive to deviate, even not the grand coalition. Theorem 3 and the techniques of its proof apply only for $\bar{N} < N - 1$. In particular, they do not map a strong equilibrium of G into a strong equilibrium in G_{async}.

4 DISCUSSION

Intuitively, we can think of the protocol constructed in the proof of Theorem 1 as consisting of two stages: The first stage is a commitment stage, where each of the players chooses a strategy and commits to it by sending its "shares" to a pair of other players. Receiving a single share from a couple of players does not, in that setting, enable a player to obtain information about the others' strategies. In the second stage, a revelation stage, players reveal their secrets and the shares they received from other players in the first stage. The secret sharing scheme we used had the property that as long as at most one player deviated from the protocol, a player could not succeed in claiming that it committed to a value that differs from the one it sent out in the first phase.

It turns out that the way players commit in the first stage is of importance and one should be careful about the requirements from such a commitment stage. The following example highlights another aspect of the desired technique.

Example 2 *Consider the IR version of the two-player game of matching pennies, with players ROW and COLUMN:*

	A	B
A	$0, 2$	$2, 0$
B	$2, 0$	$0, 2$

Assume that, by some magic way, the ROW player can commit to some strategy (without loss of generality, assume it is A) by sending some encrypted message, say $M(A)$, to the BB. This means that, by reading $M(A)$, COLUMN has no information on the strategy of ROW (i.e., he still believes ROW plays A with probability 0.5), yet ROW cannot claim to play B at the revelation stage. However, COLUMN player can now read $M(A)$ from the BB and send it as his commitment as well (note that COLUMN has committed to a strategy he does not know). Once ROW reveals his strategy A, COLUMN immediately reads that strategy and claims it as his own as well. Consequently COLUMN is always able to match ROW and win.

What the example shows is that the commitment stage must be constructed with sufficient secrecy so that opponents will obviously have no additional information and furthermore opponents will not be able to correlate their own strategy with that of the player. We note that the commitment stage used in our proofs has this property as well.[6]

5 RELATED RESEARCH

5.1 Mental Games

The problem of making a 'game' playable, in particular over a communication network, without the help of an extra player, has been raised by computer scientists for a while now. Goldreich, Micali and Wigderson [9] may be one of the earliest papers in which this issue is discussed. Moreover, Goldreich, Micali and Wigderson, and later others (e.g., Ben-Or *et al.* [5], [22] and [4]) discuss optimal play by players in synchronous, as well as asynchronous settings.

This literature, however, has a different focus than ours. Broadly speaking, its emphasis has been on devising ways in which a protocol that employs a trusted dealer can be faithfully simulated by one in which such a dealer does not exist. A main tool for this is the implementation of a joint computation by the players of an arbitrary function of their individual secrets (e.g., secret strategies) without any player obtaining any information about the secrets beyond the value of this function. In contrast, our focus has been on relating equilibria of normal form games and an analogous notion we defined for protocols. Our model is not concerned with ultimate secrecy beyond insisting that no strategically relevant information is obtained that can modify the

[6] One natural way to achieve such a commitment scheme is by requiring each player to 'prove' he knows the value he has committed to. Obviously it is important that such a proof will not disclose additional information about the secret, beyond the mere fact that the player himself actually knows the value of the secret. This is the essence of *Zero Knowledge Proofs* introduced by Goldwasser, Micali and Rackoff [11].

player's behavior. Moreover, our approach assumes that players are rational, so that they will deviate from their original strategy only if they can expect to gain from such deviation. As a result, the computations that our solutions required have much lower complexity than that typically required in the cryptography literature.

Our results make extensive use of the assumption of individual rationality of games in order to ensure the existence of an equivalence-preserving implementation of the game by distributed protocols. This allows us to motivate (or rather coerce) a rational player not to default in the midst of the execution of a protocol if it expects a negative payoff at some point. In most of the protocols considered in the cryptographic literature, there is a point in the execution after which a player's defaulting will not affect the final outcome. Moreover, before that point the player is as motivated to participate in the protocol as she is when facing a trusted dealer. However, the setting does not address the question of why, in the setting with the trusted dealer, the player would go along to begin with. We are faced with this issue because we start from the normal form game in an economically motivated setting.

5.2 Replacing Cryptography with Rationality

The idea of considering distributed protocols, whilst replacing cryptography by rationality in such protocols, taking into account the structure of communications, has been discussed in an unpublished manuscript by Monderer and Tennenholtz (see [18] for an extended abstract).

In their paper, Monderer and Tennenholtz focus on issues related to mechanism design and show that in various communication topologies (2-connected graphs and in particular rings) one can leverage on the rationality of agents and use light-weight cryptographic schemes while ensuring the execution of the distributed protocol (the mechanism) in a satisfactory way. In a related paper [19], Monderer and Tennenholtz define an intriguing model of *distributed games* that is intended to capture games played over a rich asynchronous setting such as a distributed system.

In oontrast, this paper does not deal with the issue of mechanism design. Moreover, rather than defining a new model of activity as Monderer and Tennenholtz do, we set out to relate in a more or less direct fashion well established existing models in game theory and distributed systems. Our definitions introduce notions of equilibria at the protocol level. There is undoubtedly a close connection between distributed games and standard distributed systems, the precise structure of which should be better understood with time.

5.3 Extensive Form Description of Normal Form Games

Kalai [15, 16], although with very different motivation, uses an approach related to ours. He considers the question of how robust a Nash equilibrium of a game in normal form is. An equilibrium is robust if it is guaranteed to emerge in extensive form descriptions of the game. Thus, Kalai looks at embeddings of a normal form game to an arbitrary extensive form of the same game and shows that for 'large and anonymous' games, Nash equilibrium is always mapped to an equilibrium of the extensive form game.

In our setup, a normal form game is mapped into a game involving time and turn-based playing. In an asynchronous model different behavior of nature (determining message delays, when players are able to move, etc.) results in a different game history. Nevertheless, similar to Kalai, we show that there is a way to map a Nash equilibrium into an equilibrium of the asynchronous game.

5.4 Knowledge of Payoffs

In the models we have considered so far, the payoffs, at the end of the run, depend on the history of the execution as recorded on the bulletin board. In Nissan [21] results are required to be public knowledge in a similar setting. A more general definition could have the payoffs be a function of all *external*, or *observable* events, such as the messages communicated. Thus, for example, if player 1 needs to lie according to a particular

strategy, this fact could affect the player's overall payoff. Since communication among players is in large part a private matter, such a general definition would suffer from the problem that payoffs need not necessarily be known to the players themselves. Moreover, in an asynchronous setting, without a central entity such as the bulletin board, it is *impossible* to make the outcome common knowledge, due to results of Halpern and Moses [12].

6 FUTURE RESEARCH

This paper presents a preliminary framework relating the classical game theoretic modelling with that of asynchronous distributed systems. The issues discussed here should be further extended in a variety of directions. We now consider a number of directions worthy of further investigation.

6.1 Two-Player Games and ϵ-equilibrium

The secret sharing technique we used in Section 3.1 allowed us to obtain a representation that preserves Nash equilibria for games with $N > 2$ players, but not for two-player games. In the secret sharing schemes we have considered, each player must send shares of his secret to a minimum of two other players. Hence, they are not applicable to two-player games. In fact, we assess that the two-player case is impossible.

However, there may be some good news. Assuming players are computationally limited, we can resort to well-known cryptographic algorithms (e.g., as in Goldreich [10]) which almost have the desired properties. Roughly speaking, players commit to their strategies by sending each other encrypted messages stating their choices. After both have received the encrypted messages, they each "reveal" their strategies by showing what has been encrypted and how. Under widely accepted cryptographic assumptions, such algorithms exist that are guaranteed to have the desired properties with high probability. Information is "leaked" during the commitment process only

with negligibly small probability. Consequently, the results we showed for $N > 2$ players can be extended to the two-player case provided equilibrium is replaced with the weaker notion of ϵ-equilibria. In fact, the same is true also for strong equilibria in N-player games for general N.

6.2 Dropping the IR Assumption

Throughout the paper we assumed the games to be IR. We note that this can be relaxed in two ways.

The first relaxation is straightforward. An equilibrium of a game G is called *ex-Post* IR if the payments to all players in the support of the equilibrium (i.e., in any strategy tuple which is played with positive probability) is nonnegative.

Remark: In Theorems 1–3, we may substitute the assumption that G is IR with the quantifier *for every ex-Post IR equilibrium of G*.

Consider a mixed-strategy equilibrium of a game G that may lead to negative payments, but on average leads to nonnegative payments. Such an equilibrium is called *ex-ante* IR. Can Theorems 1–3 be relaxed in order to accommodate ex-ante IR equilibria in non-IR games? We assert this is possible, assuming players are computationally limited. In this case, one can probably leverage on techniques of *verifiable secret sharing* introduced by Chor *et al.* [8] to achieve such proofs.

6.3 Subgame-Perfect Equilibrium

The implementation of a game in an asynchronous distributed system automatically induces a game that is played a long time, and so players can rethink their strategy during the course of a run if they observe that their opponents do not play as anticipated. Selten [24, 25] introduced the notion a *Subgame-Perfect Equilibrium* in models of extensive form games to capture this intuition. Two natural questions are whether 'subgame perfect equilibria' can be naturally ported to the asynchronous setup and whether Nash (or ϵ) equilibria can be mapped into such a stronger notion of equilibrium. This is a subject of further investigation.

6.4 Different Models for the Asynchronous Setup

In this work we chose one specific model for the asynchronous setup. However, there are various variations on our model.

1. **Bulletin Board** - Can one dismiss the bulletin board and make use only of the N original players? An intermediate model in this case could be one without the bulletin board but with some form of a broadcast channel that guarantees that identical messages are delivered to all players. A most likely candidate for this would involve replacing the act of posting a message to the BB by performing a so-called *Atomic broadcast* of the message to all players [17]. Notice that every attempt to replace the BB would require, in addition, an alternative definition of the basis for the payoffs in a run. In our definitions, the payoffs in a given run are determined as a function of the history of the BB.

2. **Clearing Mechanism** - In our setup the clearing mechanism is a function of the history recorded in the BB. What if we allow it to take into account additional information, as discussed above, such as the contents of the communication taking place over private channels?

3. **Private Channels** - Can our results be replicated when private channels are not fully reliable? What can be achieved in the absence of such channels?

References

[1] Aumann, R. J., "Correlated Equilibrium as an Expression of Bayesian Rationality". *Econometrica* **55**, 1987, pp. 1–18.

[2] Aumann, R. J., "Acceptable Points in General Cooperative n-Equilibrium Games". In H.W. Kuhn and R.D. Luce, eds., *Contribution to the Theory of Games IV*, Princeton University Press, 1959, pp. 287–324.

[3] Barany, I., "Fair Distribution Protocols or How the Players Replace Fortune". *Math. of Oper. Res.* **17**(2), 1992, pp. 327–340.

[4] Ben-Or, M., Canetti, R. and O. Goldreich, "Asynchronous Secure Computation". *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, 1993, pp. 52–61.

[5] Ben-Or, M., Goldwasser, S. and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (extended abstract)". *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, 1988, pp. 1–10.

[6] Blumrosen, L. and N. Nissan, "Auctions with Severely Bounded Communication". *Proc. of the 43rd Annual Symposium on Foundations of Computer Science (FOCS'02)*, 2002.

[7] Blundo, C., De Santis, A. and U. Vaccaro, "On Secret Sharing Schemes". *IPL*, 65, 1998, pp. 25–32.

[8] Chor, B., Goldwasser, S., Micali, S. and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults". *Proc. 26th Conference on Foundations of Computer Science (FOCS)*, 1985, pp. 383–395.

[9] Goldreich, O., Micali, S. and A. Wigderson, "How to Play any Mental Game". *Proc. 19th Symp. on Theory of Computing (STOC)*, 1987, pp. 218–229.

[10] Goldreich, O., *Foundations of Cryptography, Basic Tools*. Cambridge University Press, 2001.

[11] Goldwasser, S., Micali, S. and R. L. Rackoff., "The Knowledge Complexity of Interactive Proof Systems". *SIAM Journal of Computing* **18**, 1989, pp. 186–208.

[12] Halpern, J. Y. and Y. Moses, "Knowledge and Common Knowledge in a Distributed Environment". *Journal of the ACM* **37**(3), 1990, pp. 549–587.

[13] Harsanyi, J. C., "Games with Incomplete Information". *Management Science* **14**, 1967, pp. 159–182.

[14] Holzman, R., Kfir-Dahav, N., Monderer, D. and M. Tennenholtz, "Bundling Equilibrium in Combinatorial Auctions". *Games and Economic Behavior*, forthcoming (2001).

[15] Kalai, E., "Ex-Post Stability in Large Games". Center for Mathematical Studies in Economics and Mathematical Science, Discussion paper 1351, Northwestern University, 2001.

[16] Kalai, E., "Large Robust Games". Center for Mathematical Studies in Economics and Mathematical Science, Discussion paper 1350, Northwestern University, 2002.

[17] Lynch, N. A., *Distributed Algorithms*. MIT Press, 1996.

[18] Monderer, D. and M. Tennenholtz, "Distributed Games: From Mechanisms to Protocols". *Procceedings of the 16th NCAI Conference (AAAI), 1999*, pp. 32–37.

[19] Monderer, D. and M. Tennenholtz, "Distributed Games". *Games and Economic Behavior* **27**, 1999.

[20] Naor, M., "Bit Commitment Using Pseudo-Randommness". *J. Cryptology* **4**(2), 1991, pp. 151–158.

[21] Nissan, N., "Algorithms for Selfish Agents–Mechanism Design for Distributed Computation". In C. Meinet and S. Tison, eds., *Proceedings of STACS 99*, Springer-Verlag LNCS **1563**, 1999, pp. 1–15.

[22] Rabin, T. and M. Ben-Or, "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (extended abstract)". *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, 1989, pp. 73-85.

[23] Roye, P. F., "Challenges for the Mutual Fund Industry in the Competitive Frontier" (http://www.sec.gov/news/speech/spch358.htm). Speech held at the Mutual Funds and Investment Management Conference, Palm Desert, Calif. March 27, 2000.

[24] Selten, R., "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfragetrgheit -Teil I,II: Bestimmung des dynamischen Preisgleichgewichts (An

Oligopoly Model with Demand Inertia)". *Zeitschrift fr die gesamte Staatswissenschaft* **121**, 1965, 301–324 and 667–689.

[25] Selten, R., "Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games". *International Journal of Game Theory* **4**(1), 1975, pp. 25–55; reprinted in H.W. Kuhn, ed., *Classics in Game Theory*, Princeton University Press, 1997, pp. 317–354.

[26] Shamir A., "How to Share a Secret". *Communications of the ACM* **22**(11), 1979, pp. 612–613.

[27] Shoham, Y. and M. Tennenholtz. "On Rational Computability and Communication Complexity". *Games and Economic Behavior* **35**, 2001.