0368.4170: Cryptography and Game Theory	Ran Canetti and Alon Rosen
Lecture 7	
02 December 2009	Fall 2009

Scribe: R. Ring

In this lecture we will talk about

- Two-Player zero-sum games (min-max theorem)
- Mixed strategy Nash equilibrium (existence, computational complexity)
- ε-NE
- Correlated equilibrium
- How to use cryptography to implement the correlated equilibrium
  - Computational Game

1 Two-Player Zero-Sum Games

1.1 Zero-Sum games with pure strategies

Definition 1.1 (Zero-Sum game)**.** *$G = (N, (A_i), (u_i))$ is a zero-sum game if:*

$$\forall (a_1, ..., a_n) \in A_1 \times ... \times A_n \ \sum_{i \in N} u_i(a_1, ..., a_n) = 0 \, .$$

A game is considered zero-sum if the sum of the payoffs of all the players is zero for any choice of the strategies. We focus on two-player zero-sum game where n=2, $(a_1, a_2) \in A_1 \times A_2$ and $u_1(a_1, a_2) + u_2(a_1, a_2) = 0$. Zero-sum games are a special case of "constant sum games". The latter have a similar definition, but the sum of payoffs can be any constant (and not only zero). Usually the sum is normalized so as to set the constant to zero. In a two-player zero-sum game, when a player tries to maximize his payoff, he simultaneously minimizes the payoff of the other player.

Example: Matching Pennies

Consider a standard Matching Pennies game, whose payoff is given by:

	Head	Tail
Head	1, -1	-1, 1
Tail	-1, 1	1, -1

Evidently, this is a two-player zero-sum game, because the sum of the utilities in each entry of the payoff matrix is zero.

We are interested in solution concepts for zero-sum games. A convenient way to reason about these kind of games is to consider an interactive two stage game where one player acts first and the other player acts second; as always, each of the players tries to optimize his own payoff.

- P_1 chooses an action $a_1 \in A_1$;
- P_2 chooses the best response $a_2 \in A_2$;
- P_1 gets the $\min_{a_2 \in A_2} u_1(a_1, a_2)$.

Let's suppose that P_1 is the row player and P_2 is the column player. The best strategy for P_1 is a (pure) strategy $\max_{a_1 \in A_1}\{\min_{a_2 \in A_2}\{u_1(a_1,a_2)\}\}$, and he is called the *maximizer*. If P_1 goes second, then the best possible utility payoff would be $\min_{a_2 \in A_2}\{\max_{a_1 \in A_1}\{u_1(a_1,a_2)\}\}$ and P_1 would be then called the *minimizer.* Note that here we assume that the game is risk neutral, which means that a player only cares about maximizing the expectation value of his own gain.

In the Matching Pennies example if P_1 goes first, then he gets -1; and if P_1 goes second, he gets 1. Thus there is a big advantage to play second, and it is not clear how to play if the actions are taken simultaneously.

1.2 Zero-Sum games with mixed strategies

Recall that mixed strategy s_i for player i is defined as an element of $\Delta(A_i)$ where

- $\Delta(A_i)$ is the set of all probability distributions over A_i,
- $s_i(x) = Pr[s_i = x]$,
- $\text{supp}(s_i) =$ set of all $a_i \in A_i$ such that $s_i(a_i) > 0$.

In words: a mixed strategy $s_i(x)$ is the probability that player i will use his pure strategy x. The support of a mixed strategy s_i is the set of all different pure strategies that are used with non-zero probability.

Mixed strategies are used to get a relaxation for dominant or Nash equilibrium.

Notation: Throughout this course the *expected utility* of a game is denoted $U_i(s_i, s_{-i})$ (strictly speaking it should be denoted $E(u_i(s_i, s_{-i}))$).

Let v_i be the maximum over all mixed strategies for player i of the minimum over all mixed strategies of the other player. That is,

$$v_i = \max_{s_i \in \Delta(A_i)} \{ \min_{s_{-i} \in \Delta(A_i)} \{U_i(s_i, s_{-i})\}\} .$$

Also define

$$\bar{v}_i = \min_{s_{-i} \in \Delta(A_i)} \{ \max_{s_i \in \Delta(A_i)} \{U_i(s_i, s_{-i})\}\} .$$

Observation 1. $v_i \leq \bar{v}_i = -v_{-i}$.

Proof. We start by proving that $v_i \leq \bar{v}_i$:

$$\forall s_i \in \Delta(A_i), \min_{s_{-i} \in \Delta(A_i)} \{U_i(s_i, s_{-i})\} \leq U_i(s_i, s_{-i})$$

hence,

$$\max_{s_i \in \Delta(A_i)} \{ \min_{s_{-i} \in \Delta(A_i)} \{U_i(s_i, s_{-i})\}\} \leq \max_{s_i \in \Delta(A_i)} \{U_i(s_i, s_{-i})\} .$$

Using this inequality with the particular value of $s_i \in \Delta(A_i)$, which minimizes the right hand side, one obtains the desired inequality.

Now we prove that $\bar{v}_i = -v_{-i}$.

By definition in a zero-sum game $U_i(s_i, s_{-i}) = -U_{-i}(s_i, s_{-i})$, hence

$$v_i = \max_{s_i}\{\min_{s_{-i}}\{U_i(s_i, s_{-i})\}\} = \max_{s_i}\{\min_{s_{-i}}\{-U_{-i}(s_i, s_{-i})\}\} .$$

By applying the obvious relations $\max(-f(x)) = -\min(f(x))$ and $\min(-f(x)) = -\max(f(x))$ one obtains $-\min_{s_i}\{\max_{s_{-i}}\{U_{-i}(s_i, s_{-i})\}\} = -\bar{v}_i$. □

The above results show formally that it is better to go second and that the minimizing strategy of player i equals minus the maximizing strategy of the other player.

Theorem 1.1 (MinMax, Von Neumann (1928))**.** *For any finite two-player zero-sum game and any* $i \in 1, 2$*:*

$$\max_{s_i}\{\min_{s_{-i}}\{U_i(s_i, s_{-i})\}\} = \min_{s_{-i}}\{\max_{s_i}\{U_i(s_i, s_{-i})\}\}$$

This theorem can be proved by Linear Programming methods. It demonstrates the usefulness of using mixed strategies (the MinMax strategy is a mixed strategy). As a bonus it gives an efficient way to play the game and guarantee the value v_i. Using the previously defined notation, the theorem can be written as $v_i = \bar{v}_i$ or $v_1 + v_2 = 0$, that is, for zero-sum games it does not matter whether you play first or second. In addition, players do not necessarily need to know what is the strategy of the other party; it is enough for each player to solve his own optimization problem.

The usefulness of the MinMax theorem can be exemplified in the Matching Pennies game. As we have seen before, the Matching Pennies game has neither a dominant strategy nor a NE. However, the strategy where both players are randomizing between "Tail" and "Head" with equal probability of $\frac{1}{2}$ guarantees that the expected utility of each of the players will be zero. This strategy is both the MaxMin strategy and the MinMax strategy for each player and is also a dominant mixed strategy.

2 Mixed Nash Equilibrium

The notion of a *mixed strategy Nash equilibrium* is designed to model a steady state of a game in which the participant's choices are not deterministic but are regulated by probabilistic rules.

Definition 2.1 (Mixed Nash Equilibrium)**.** *A vector* $s = (s_1, ..., s_n)$ *of mixed strategies in a game* $G = (N, (A_i), (u_i))$ *is said to be in NE if for each player* $i \in N$*, and for each mixed strategy* $s_i' \in \Delta(A_i)$*,* $s_i' \neq s_i$*, the following holds:*

$$U_i(s_i, s_{-i}) \geq U_i(s_i', s_{-i}) .$$

In case of a strict inequality the equilibrium is a strict NE.

In other words, given that all the other players play according to s, it does not pay off for player i to deviate. Alternatively, every pure strategy in $supp(s_i)$ is the best response to s_{-i}.

Example: Bach & Stravinsky

	Bach	Stravinsky
Bach	2, 1	0, 0
Stravinsky	0, 0	1, 2

As we saw in the previous lecture this game has 2 pure NE: (Bach, Bach) and (Stravinsky, Stravinsky). A mixed NE of this game is $(\frac{2}{3}, \frac{1}{3})(\frac{1}{3}, \frac{2}{3})$ with expected payoff $\frac{2}{3}$.

Theorem 2.1 (Nash (1951))**.** *Every finite strategic game has a mixed strategy NE.*

The theorem relies on Brower's fix points theorem. A mixed NE can be computed by the Lemke-Howson algorithm (1964). Throughout this course we will consider games which have Nash equilibria that we design into the game and so will not be concerned with their existence.

Proposition 2.1. *$s = (s_1, ..., s_n)$ is a mixed NE in $G(N, (A_i), (u_i))$ iff:*

1. $U_i(a_i, s_{-i}) = U_i(a_i', s_{-i}), \forall a_i, a_i' \in supp(s_i)$
2. $U_i(a_i, s_{-i}) \geq U_i(a_i', s_{-i}), \forall a_i \in supp(s_i), \forall a_i' \notin supp(s_i)$.

In other words, for any two strategies in the support of the NE the expected utility is the same. Also, when playing in the support of NE, the expected utility is equal to or greater than that gained when playing outside the support. The proposition gives additional information about the pure strategies. Determining the support usually turns out to be a complicated problem.

Proof. $\Rightarrow$ suppose $s = (s_1, ..., s_n)$ is a NE and suppose that (1) or (2) does not hold. Then if (1) does not hold then there exist a pair of actions a_i, a_i' in support such that $U_i(a_i', s_{-i}) > U_i(a_i, s_{-i})$ in contradiction to s being NE. If (2) does not hold, then there exists a pair of actions a_i, a_i' such that $a_i' \notin supp(s_i)$ and $a_i \in supp(s_i)$ and $U_i(a_i', s_{-i}) > U_i(a_i, s_{-i})$. Now we change the s_i in the following way: whenever we are supposed to play a_i (and we do so because $a_i \in supp(s_i)$), we play a_i'. Then we'll get a strictly better expected payoff in contradiction to s being a NE.

$\Leftarrow$ suppose (1) and (2) hold but s is not a NE. Then $\exists i \in N$ such that $U_i(s_i', s_{-i}) > U_i(s_i, s_{-i})$. In particular $\exists a_i' \in supp(s_i')$ such that $U_i(a_i', s_i') > U_i(s_i, s_{-i})$. If (1) holds then $a_i' \notin supp(s_i)$ must hold and this contradicts (2). □

A corollary of the above (which we will not prove) is that given $supp(s_i) \forall i \in N$ where s is a NE, it is possible to compute a NE in polynomial time.

The definitions introduced so far have several drawbacks. They do not deal with situations where multiple equilibria exist. If the players fail to choose the same equilibrium, they may not reach one at all. This situation may arise, for instance, if the players are indifferent to all the strategies s whose supports are subsets of their equilibrium strategies. This problem can be avoided by designing a game with a single equilibrium. Moreover, the requirement that strategies are played independently of each other is a very restrictive one, and may not be satisfied in reality. Another simplifying assumption is simultaneity, which effectively eliminates interactions between the players. Lastly, a mixed NE may be hard to compute. The complexity question has two flavors: 1) given a game, find a mixed NE effectively; 2) given a mixed NE, sample the distribution effectively. Note that in this course we will not deal with the complexity of playing a NE.

2.1 The complexity of finding a NE for two-player games

A pure dominant strategy, a pure NE and a MinMax strategy can all be found in polynomial time (if they exist). A pure dominant strategy can be found by checking each pair of strategies for being a dominant strategy. A MinMax strategy of a zero-sum game can be found via Linear Programming.

In general, a two-player mixed NE can be found in exponential time by using the Lemke-Howson algorithm (Lemke & Howson, 1964). The correctness of the Lemke-Howson algorithm provides an alternative proof to the existence of two player NE in any game. It resembles the Linear Programming Simplex algorithm; its main idea is the following. The algorithm maintains a single guess as to what the supports should be, and in each iteration changes the guess a little bit and solves a linear programing problem. The algorithm can be viewed as a walk along the edges of an n-dimensional

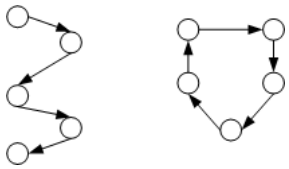

Figure 1: A typical problem in PPAD.

*polytop*e (a polytope is the set of solutions of a system of linear inequalities $mx \leq b$ where m is a real matrix and b is a real vector). The following two cases of finding a two-player mixed NE are NP-hard: finding a NE where the column player has an expected positive payoff and finding a NE where the row player has a mixed strategy. Some of these problems can be shown to be NP-hard via a reduction from the CLIQUE or SAT problems.

Some evidence of the hardness of finding two-player NE come from the fact that it is PPAD complete. A typical problem in PPAD (Polynomial Parity Argument Directed, Papadimitriou (1994)) is defined as follows. The input is a directed graph with exponential number of vertices $v \equiv \{v_i\}$ and one known source vertex (a vertex with no incoming edges) called the "standard source". The graph has no isolated nodes and the in-degree and out-degree of each vertex is at most one. Information about the graph is explicitly given via a polynomial-time computable function $f(v_i)$ (polynomial in the size of v) which returns the incoming and the outgoing edges of v_i (if they exist). The problem is to find a sink (a vertex with no outgoing edges), or any source other than the standard one. This is a "search" problem, and it is always guaranteed that a solution exists. This is true because for every directed graph with in/out-degree at most one without isolated nodes, every node is either part of a cycle or part of the path from a source to sink, then for a given source node there exists a sink node that is either a descendant of the source or the source itself.

3 ε-NE

Computing a mixed NE is a difficult problem, thus computing an approximate equilibrium seems to be a very attractive compromise. Whereas in mixed NE it does not pay off for a player to deviate from his strategy, here it may pay off, but not more than ε.

Definition 3.1 (ε-Nash Equilibrium)**.** *Let $\varepsilon \geq 0$. A vector $s = (s_1, ..., s_n)$ is said to be in ε-NE if for any mixed strategy $s'_i \in \Delta(A_i)$, $\forall i \in N$, the following holds:*

$$U_i(s_i, s_{-i}) \geq U_i(s'_i, s_{-i}) - \varepsilon \ .$$

A possible (stronger) alternative way of defining ε-NE is as follows: with expectation over strategy, the gain is no more than ε, and with probability $\geq 1 - \varepsilon$ there is no gain at all. Note that for $\varepsilon = 0$ one obtains a mixed NE, and for $\varepsilon < 0$ one obtains a strict NE. The following example exemplifies why ε-NE may be problematic.

Example:

	Left	Right
Up	1, 1	-100, 1+ε
Down	1+ε, -100	-99, -99

The strategy (Up, Left) is ε-NE: assuming that the column player sticks to Left, then by deviating to Down the row player can not get more then ε; assuming that the row player sticks to Up, the column player will get no more then ε by deviating to Right. The problem is to answer the question: "Why not to go to 1+ε?". A possible answer is: when the players start to deviate from their strategies, the meaning of the steady state (mixed NE) is lost.

We can justify lack of deviation from NE by arguing that a deviation costs something. Consider the following example: some organization has implemented a cryptographic protocol and deployed it to all its customers. Now the customers should decide whether to deviate or not. After analyzing the situation it has been found that the current state is an ε-NE, and the customers have been shown that even if they deviate they can not gain more then ε. However, changing the software does costs something (time, money, etc.), thus the customers may not be interested in deviating under the conditions above.

3.1 Complexity of ε-NE

The latest reductions show that it is PPAD complete to compute ε-NE for polynomially small ε. Also there are results showing that for any NE and for any $\varepsilon > 0$ there is a "nearby" ε-NE with small support size $O(\frac{\log(n)}{e^2})$, where n is number of pure strategies. The intuition behind those results is that any distribution can be approximated by a small number of samples. Any NE is a distribution, so by picking randomly from the support one can obtain a good approximation (i.e. an approximation having an expected utility close to the real one) for the distribution.

Corollary 3.1. *One can optimize with $n^{O(\log(n)/e^2)}$ time to find ε-NE.*

4 Correlated Equilibrium

Hitherto we have made an assumption that actions are taken independently. Consider, for example, the following variation of the Bach & Stravinsky game, when one of the concerts takes place in a closed hall and the other in the park. If it is raining when the players are making their decisions, this external information may influence their choices. Thus we have a bias that should be taken into account. Another example is a situation where it is hard to find a NE. Here one cannot guarantee that the players will play at all. In general, the existence of a third party (a random event that can be observed like sunspots, weather, traffic lights) may help the players to achieve a better outcome.

Example: Chicken Game

	Stop	Go
Stop	0, 0	0, 1
Go	1, 0	-1, -1

Here the payoffs are supposed to represent a situation in which two drivers speed up toward a junction. Each driver has 2 options: Stop or Go. There are 2 pure NE: (Stop,Go) and (Go, Stop) and one mixed NE $(\frac{1}{2}, \frac{1}{2})$. Here traffic lights can help the players to choose a strategy that will yield a positive payoff for both players. This motivates the following definition.

Definition 4.1 (Mediated Game)**.** *A mediated version of a game $G = (N, (A_i), (u_i))$ is a game which consists of two stages:*

- ***Stage 1**: A mediator chooses a vector of actions $a = (a_1, ..., a_n) \in A_1 \times, ..., \times A_n$ according to some distribution M. Mediator hands a_i to player i.*
- ***Stage 2**: Players play the game G.*

The actions a_i are a set of recommendations, which the players may follow or not. In the previous example the traffic lights played the role of a mediator and the recommendations were green and red meaning "go" or "stop", respectively.

Let $U_i(a_i', a_{-i} \,|\, a_i)$ denote the expected utility of player $i \in N$, given that he plays a_i' after having received a recommendation a_i and all other players play a_{-i}.

Definition 4.1 (Correlated Equilibrium, Aumann (1956))**.** *A distribution $M \in \Delta(A)$ is a correlated equilibrium in a game if $\forall a \in supp(M)$, $\forall i \in N$ and $\forall a_i' \in A_i$, the following holds:*

$$U_i(a_i', a_{-i} \,|\, a_i) < U(a_i, a_{-i} \,|\, a_i) \; .$$

Recommendations are correlated. Thus given his recommendation, a player has some information about the recommendation of the other players. Hence for any possible outcome of the mediator, it does not pay off for a player to deviate from his strategy.

Any mixed NE is a correlated equilibrium. Specifically, a mixed NE is a *product* distribution over $A_1 \times A_2 \times ... \times A_n$ whereas correlated equilibrium is an *arbitrary* distribution over $A_1 \times A_2 \times ... \times A_n$.

A correlated equilibrium can be computed in polynomial time because the distribution M can be found via linear programming.

In some cases a correlated equilibrium gives a better payoff for both players than any mixed NE.

Example:

	Left	Middle	Right
Up	2, 1	1, 2	0, 0
Middle	0, 0	2, 1	1, 2
Down	1, 2	0, 0	2, 1

In this two-player game each player has 3 possible actions as follows from the table. A pure NE for this game does not exist. There is a unique mixed NE $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, whose expected payoff is 1. A correlated equilibrium is obtained by playing each non-zero cell with probability $\frac{1}{6}$. The latter can be achieved by adding a mediator that chooses one of the options (Up, Left), (Up, Middle), (Middle, Middle), (Middle, Right), (Down, Left), (Down, Right) with probability $\frac{1}{6}$. Thus, for example, if the row player is given the Up option then he knows that the column player is given the Middle or Left options. The expected payoff for the correlated equilibrium is 1.5. This game is an example of a game whose correlated equilibrium pays off more than its unique mixed NE.

5 Implementing Correlated Equilibrium by Cryptography

As we have seen in the previous section, if the playing parties assume the existence of a trusted mediator, then they can potentially achieve a correlated equilibrium that may be "preferable" to any of the available Nash equilibria. If a trusted mediator is not available, the question is how can the

parties themselves run a protocol in place of the mediator? This question was first explored in the economics community, where researchers suggested “cheap talk" protocols by which parties could communicate amongst themselves to implement a correlated equilibrium. The communication among the players is “cheap" in the sense that it costs nothing; it is also “worth nothing" in the sense that players are not “bound” to any statements they make; e.g., there is no legal recourse if someone lies.

One natural way to implement such a protocol would be by using secure computation in the following way (we consider only two-player games):

1. Given a game G, a correlated equilibrium should be found. This can be done in polynomial time by finding a distribution sampled by M (usually it will be a randomized function).
2. Implement M using a two-party computation protocol Π. The protocol should be implemented correctly, fairly and be private.
3. The resultant new game G' includes the Π's messages as actions. The expected payoffs of G' in equilibrium corresponds to the payoffs of the correlated equilibrium.

A correlated equilibrium implemented using cryptography can be computed in polynomial time, and is guaranteed to be as good as the best NE.

These definitions introduce several issues that should be addressed. First, one has to define a notion of *computational NE* where the strategies of both players are restricted to probabilistic polynomial time. Since a computational model is considered, the definitions must account for the fact that the players may break the underlying cryptographic scheme with negligible probability, thus gaining some advantage in the game (here ε-NE will be of help). Moreover, hitherto the game has been considered as a fixed object where no parameter can diverge, but now, using a computational model, the asymptotics has to be handled. Another important issue related to computability is the usage of a sequence of games, to be discussed in the next lectures..

Second, in order to find a correlated equilibrium (the first step) one needs to use the notion of interactive games whose actions are messages. The definition and example of interactive game will be introduced in the next lecture.

References

[1] Noam Nisan et al. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[2] J. Katz. Bridging game theory and cryptography: Recent results and future directions. *5th Theory of Cryptography Conference (TCC), Springer-Verlag (LNCS 4948)*, pages 251–272, 2008.

[3] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.