A Cryptographic Solution to a Game Theoretic Problem

Yevgeniy Dodis[1], Shai Halevi[2], and Tal Rabin[2]

[1] Laboratory for Computer Science, MIT, 545 Tech Square, Cambridge, MA 02139, USA. Email: yevgen@theory.lcs.mit.edu.
[2] IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598, USA. Email: {shaih,talr}@watson.ibm.com.

Abstract. In this work we use cryptography to solve a game-theoretic problem which arises naturally in the area of two party strategic games. The standard game-theoretic solution concept for such games is that of an *equilibrium*, which is a pair of "self-enforcing" strategies making each player's strategy an optimal response to the other player's strategy. It is known that for many games the expected equilibrium payoffs can be much higher when a trusted third party (a "mediator") assists the players in choosing their moves (*correlated equilibria*), than when each player has to choose its move on its own (*Nash equilibria*). It is natural to ask whether there exists a mechanism that eliminates the need for the mediator yet allows the players to maintain the high payoffs offered by mediator-assisted strategies. We answer this question affirmatively provided the players are computationally bounded and can have free communication (so-called "cheap talk") prior to playing the game.
The main building block of our solution is an efficient cryptographic protocol to the following *Correlated Element Selection* problem, which is of independent interest. Both Alice and Bob know a list of pairs $(a_1, b_1) \dots (a_n, b_n)$ (possibly with repetitions), and they want to pick a *random* index i such that Alice learns only a_i and Bob learns only b_i. Our solution to this problem has constant number of rounds, negligible error probability, and uses only very simple zero-knowledge proofs. We then show how to incorporate our *cryptographic* protocol back into a *game-theoretic* setting, which highlights some interesting parallels between cryptographic protocols and extensive form games.

1 Introduction

The research areas of Game Theory and Cryptography are both extensively studied fields with many problems and solutions. Yet, the cross-over between them is surprisingly small: very rarely are tools from one area borrowed to address problems in the other. Some examples of using game-theoretic concepts to solve cryptographic problems include the works of Fischer and Wright [17] and Kilian [26]. In this paper we show an example in the other direction of how cryptographic tools can be used to address a natural problem in the Game Theory world.

1.1 Two Player Strategic Games

The game-theoretic problem that we consider in this work belongs to the general area of *two player strategic games*, which is an important field in Game Theory (see [20, 32]). In the most basic notion of a two player game, there are two players, each with a set of possible moves. The game itself consists of each player choosing a move from its set, and then both players executing their moves simultaneously. The rules of the game specify a *payoff* function for each player, which is computed on the two moves. Thus, the payoff of each player depends both on its move and the move of the other player. A *strategy* for a player is a (possibly randomized) method for choosing its move. A fundamental assumption of these games, is that each player is *rational*, i.e. its sole objective is to maximize its (expected) payoff.

A pair of players' strategies achieves an *equilibrium* when these strategies are *self-enforcing*, i.e. each player's strategy is an *optimal response* to the other player's strategy. In other words, once a player has chosen a move and believes that the other player will follow its strategy, its (expected) payoff will not increase by changing this move. This notion was introduced in the classical work of Nash [31].

In a *Nash equilibrium*, each player chooses its move *independently* of the other player. (Hence, the induced distribution over the pairs of moves is a product distribution.) Yet, Aumann [2] showed that in many games, the players can achieve much higher expected payoffs, while preserving the "self-enforcement" property, if their strategies are *correlated* (so the induced distribution over the pairs of moves is no longer a product distribution). To actually implement such a *correlated equilibrium*, a "trusted third party" (called a *mediator*) is postulated. This mediator chooses the pair of moves according to the right joint distribution and *privately* tells each player what its designated move is. Since the strategies are correlated, the move of one player typically carries some information (not known a-priori) on the move of the other player. In a correlated equilibrium, no player has an incentive to deviate from its designated move, even knowing this extra information about the other player's move.

1.2 Removing the Mediator

As the game was intended for two players, it is natural to ask if correlated equilibria can be implemented without actually having a mediator. In the language of cryptography, we ask if we can design a two party game to eliminate the trusted third party from the original game. It is well known that in the standard cryptographic models the answer is positive, provided that the two players can interact, that they are computationally bounded, and assuming some standard hardness assumptions ([22, 34]). We show that this positive answer can be carried over to the Game Theory model as well. Specifically, we consider an *extended game*, in which the players first exchange messages (this part is called "cheap talk" by game theorists and is quite standard; see Myerson [30] for survey), and then choose their moves and execute them simultaneously as in the original game.

The payoffs are still computed as a function of the moves, according to the same payoff function as in the original game.

It is very easy to see that every Nash equilibrium payoff of the extended game is also a correlated equilibrium payoff of the original game (the mediator can simulate the pre-play communication stage). Our hope would be to show that any Correlated equilibrium payoffs of the original game can always be achieved by some Nash equilibrium of the extended game. However, Barany [3] showed that this is generally not true. Namely, that Nash equilibria payoffs of the extended game are inside the convex hull of the Nash equilibria payoffs of the original game, which often does not include many correlated equilibria payoffs of the original game (see Section 2 for an example).

In this work we overcome this difficulty by considering the realistic scenario where the players are *computationally bounded*. In other words, while Game Theory typically assumes that the players have unlimited computational capabilites when they need to make their decisions, we will assume that the players are restricted to *probabilistic polynomial time*. Of independent interest to Game Theory, we will define a new concept of a *computational Nash equilibrium* as a pair of efficient strategies where no *polynomially bounded* player can gain a non-negligible advantage by not following its strategy (see Section 3 for formal definitions). Then, we prove the following:

Theorem 1. *Let G be any two player strategic game and let G' be the extended game of G. If secure two-party protocols exist for non-trivial functions, then for any correlated equilibrium s of G there exists a computational Nash equilibrium σ of G', such that the payoffs for both players are the same in σ and s.*

In other words, any correlated equilibrium payoffs of G can be achieved using a computational Nash equilibrium of G'. Thus, the mediator can be eliminated if the players are computationally bounded and can communicate prior to the game.

We stress that although this theorem seem quite natural and almost trivial from a cryptography point of view, the models of Game Theory and Cryptography are significantly different, and thus proving it in the Game Theory framework requires some care. In particular, two-party cryptographic protocols always assume that at least one player is honest, while the other player could be arbitrarily malicious. In the game-theoretic setting, on the other hand, *both players are selfish and rational*: they (certainly) deviate from the protocol if they benefit from it, and (can be assumed to) follow their protocol otherwise. Also, it is important to realize that in this setting we cannot use cryptography to "enforce" honest behavior. This is due to the fact that a "cheating player" who was "caught cheating" during the protocol, can still choose a move that would maximizes its profit. We discuss these and some other related issues further in Section 2.

1.3 Doing it Efficiently

Although the assumption of Theorem 1 can be proven using tools of generic two-party computations [22, 34], it would be nice to obtain computational Nash equilibria (i.e. protocols) which are more efficient than the generic ones. In Section 4 we observe that for many cases, the underlying cryptographic problem reduces to a problem which we call *Correlated Element Selection*. We believe that this natural problem has other cryptographic application and is of independent interest. In this problem, two players, A and B, know a list of pairs $(a_1, b_1), \ldots, (a_n, b_n)$ (maybe with repetitions), and they need to jointly choose a random index i, so that player A only learns the value a_i and player B only learns the value b_i.[1] Our final protocol for this problem is very intuitive, has constant number of rounds, negligible error probability, and uses only very simple zero-knowledge proofs.

Our protocol for Correlated Element Selection uses as a tool a useful primitive which we call *blindable encryption* (which can be viewed as a counterpart of blindable signatures [10]). Stated roughly, blindable encryption is the following: given an encryption c of an (unknown) message m, and an additional message m', a random encryption of $m + m'$ can be easily computed. This should be done without knowing m or the secret key. Examples of semantically secure blindable encryption schemes (under appropriate assumptions) include Goldwasser-Micali [23], ElGamal [15] and Benaloh [5]. (In fact, for our Correlated Element Selection protocol, it is sufficient to use a weaker notion of blindability, such as the one in [33].) Aside from our main application, we also observe that blindable encryption appears to be a very convenient tool for devising efficient two-party protocols and suggest that it might be used more often. (For example, in the full version of this paper we show a very simple protocol to achieve 1-*out-of-n Oblivious Transfer* protocol from any secure blindable encryption scheme.)

1.4 Related Work

Game Theory. Realizing the advantages of removing the mediator, various papers in the Game Theory community have been published to try and achieve this goal. Similarly to our work, Barany [3] shows that the mediator can be replaced by pre-play communication but he requires four or more players for this communication, even for a game which is intended for two players. In his protocol only two players participate as “decision makers” during the pre-play communication, and (at least two) other players help them to hide information from each other (as Barany showed, two players do not suffice). Barany's protocol works in an information-theoretic setting (which explains the need for four players; see [6].) Of course, if one is willing to use a group of players to simulate the mediator, then the general multiparty computation tools (e.g. [6, 11]) can

[1] A special case of Correlated Element Selection when $a_i = b_i$ is just the standard *coin-flipping* problem [7]. However, this is a degenerate case of the problem, since it requires no secrecy. In particular, none of the previous coin-flipping protocols seem to extend to solve our problem.

also be used, even though the solution of [3] is simpler. Forges [18, 19] extends these results to more general classes of games. The work of Lehrer and Sorin [27] describes protocols that "reduce" the role of the mediator (the mediator receives private signals from the players and makes deterministic *public* announcements). Mailath et al. [29] show that the set of correlated equilibria of the original game coincides with the set of Nash equilibria of the so called "local-interaction game" (where many players are paired up randomly and play the original game). The distinguishing feature of our work is the observation that placing realistic computational restrictions on the players allows them to achieve results which are *provably* impossible when the players are computationally unbounded.

Cryptography. We already mentioned the relation of our work to generic two-party secure computations [22, 34]. We note that some of our techniques (in particular, the zero-knowledge proofs) are similar to those used for mixing networks (see [1, 25] and the references therein), even though our usage and motivation are quite different. Additionally, encryption schemes with various "blinding properties" were used for many different purposes, including among others for secure storage [21], and secure circuit evaluations [33].

2 Background in Game Theory

Two-player Games. Although our results apply to a much larger class of two-player games, we demonstrate them on the simplest possible class of finite *strategic games* (with complete information). Such a game G has two players 1 and 2, each of whom has a finite set A_i of possible *actions* and a *payoff function* $u_i : A_1 \times A_2 \mapsto R$ $(i = 1, 2)$, known to both players. The players move simultaneously, each choosing an action $a_i \in A_i$. The *payoff* of player i is $u_i(a_1, a_2)$. The (probabilistic) algorithm that tells player i which action to take is called its *strategy*, and a pair of strategies is called a *strategy profile*. In our case, a strategy s_i of player i is simply a probability distribution over its actions A_i, and a strategy profile $s = (s_1, s_2)$ is a probability distribution over $A_1 \times A_2$. Classical Game Theory assumes that each player is *selfish and rational*, i.e. only cares about maximizing its (expected) payoff. As a result, we are interested in strategy profiles that are *self-enforcing*. In other words, even knowing the strategy of the other player, each player still has no incentive to deviate from its own strategy. Such a strategy profile is called an *equilibrium*.

Nash equilibrium. This is the best known notion of an equilibrium [31]. It corresponds to a strategy profile in which players' strategies are *independent*. More precisely, the induced distribution over the pairs of actions, must be a product distribution, $s(A_1 \times A_2) = s_1(A_1) \times s_2(A_2)$. Deterministic (or *pure*) strategies are a special case of such strategies, where s_i assigns probability 1 to some action. For strategies s_1 and s_2, we denote by $u_i(s_1, s_2)$ the *expected* payoff for player i when players independently follow s_1 and s_2.

Definition 1. *A* Nash equilibrium *of a game G is an independent strategy profile (s_1^*, s_2^*), such that for any $a_1 \in A_1$, $a_2 \in A_2$, we have $u_1(s_1^*, s_2^*) \geq u_1(a_1, s_2^*)$ and $u_2(s_1^*, s_2^*) \geq u_2(s_1^*, a_2)$.*

In other words, given that player 2 follows s_2^*, s_1^* is an optimal response of player 1 and vice versa.

Correlated equilibrium. While Nash equilibrium is quite a natural and appealing notion (since players can follow their strategies independently of each other), one can wonder if it is possible to achieve higher expected payoffs if one allows *correlated* strategies.

In a correlated strategy profile [2], the induced distribution over $A_1 \times A_2$ can be an arbitrary distribution, not necessarily a product distribution. This can be implemented by having a trusted party (called *mediator*) sample a pair of actions (a_1, a_2) according to some *joint* probability distribution $s(A_1 \times A_2)$, and "recommend" the action a_i to player i. We stress that knowing a_i, player i now knows a *conditional distribution* over the actions of the other player (which can be different for different a_i's), but knows *nothing more*. We denote these distributions by $s_2(\cdot \mid a_1)$ and $s_1(\cdot \mid a_2)$.

For any $a_1' \in A_1, a_2' \in A_2$, let $u_1(a_1', s_2 \mid a_1)$ be the expected value of $u_1(a_1', a_2)$ when a_2 is distributed according to $s_2(\cdot \mid a_1)$ (similarly for $u_2(s_1, a_2' \mid a_2)$). In other words, $u_1(a_1', s_2 \mid a_1)$ measures the expected payoff of player 1 if his recommended action was a_1 (thus, a_2 is distributed according to $s_2(\cdot \mid a_1)$), but it decided to play a_1' instead. As before, we let $u_i(s)$ be the expected value of $u_i(a_1, a_2)$ when (a_1, a_2) are drawn according to s. Similarly to Nash equilibrium, a more general notion of a *correlated equilibrium* is defined, which ensures that players have no incentive to deviate from the "recommendation" they got from the mediator.

Definition 2. *A* correlated equilibrium *is a strategy profile $s^* = s^*(A_1 \times A_2) = (s_1^*, s_2^*)$, such that for any (a_1^*, a_2^*) in the support of s^*, any $a_1 \in A_1$ and $a_2 \in A_2$, we have $u_1(a_1^*, s_2^* \mid a_1^*) \geq u_1(a_1, s_2^* \mid a_1^*)$ and $u_2(s_1^*, a_2^* \mid a_2^*) \geq u_2(s_1^*, a_2 \mid a_2^*)$.*

Given Nash (resp. Correlated) equilibrium (s_1^*, s_2^*), we say that (s_1^*, s_2^*) achieves *Nash (resp. Correlated) equilibrium payoffs* $[u_1(s_1^*, s_2^*), u_2(s_1^*, s_2^*)]$.

Correlated equilibria of any game form a convex set, and therefore always include the convex hull of Nash equilibria. However, it is well known that correlated equilibria can give equilibrium payoffs *outside* (and significantly better!) than anything in the convex hull of Nash equilibria payoffs. This is demonstrated in the following simple example first observed by Aumann [2], who also defined the notion of correlated equilibrium. Much more dramatic examples can be shown in larger games.[2]

[2] For example, there are games with a unique Nash equilibrium s and many Correlated equilibria giving *both* players much higher payoffs than s.

	C	D
C	4,4	1,5
D	5,1	0,0

"Chicken"

	C	D
C	1/4	1/4
D	1/4	1/4

Mixed Nash s^3

	C	D
C	1/3	1/3
D	1/3	0

Correlated s^*

Game of "Chicken". We consider a simple 2 × 2 game, the so-called game of "Chicken" shown in the table to the right. Here each player can either "dare" (D) or "chicken out" (C). The combination (D, D) has a devastating effect on both players (payoffs $[0, 0]$), (C, C) is quite good (payoffs $[4, 4]$), while each player would ideally prefer to dare while the other chickens-out (giving him 5 and the opponent 1). While the "wisest" pair of actions is (C, C), this is not a Nash equilibrium, since both players are willing to deviate to D (believing that the other player will stay at C). The game is easily seen to have three Nash equilibria: $s^1 = (D, C)$, $s^2 = (C, D)$ and $s^3 = (\frac{1}{2} \cdot D + \frac{1}{2} \cdot C,\ \frac{1}{2} \cdot D + \frac{1}{2} \cdot C)$. The respective Nash equilibrium payoffs are $[5, 1]$, $[1, 5]$ and $[\frac{5}{2}, \frac{5}{2}]$. We see that the first two pure strategy Nash equilibria are "unfair", while the last mixed equilibrium has small payoffs, since the mutually undesirable outcome (D, D) happens with non-zero probability $\frac{1}{4}$ in the product distribution. The best "fair" strategy profile in the convex hull of the Nash equilibria is the combination $\frac{1}{2}s^1 + \frac{1}{2}s^2 = (\frac{1}{2}(C, D) + \frac{1}{2}(D, C))$, yielding payoffs $[3, 3]$. On the other hand, the profile $s^* = (\frac{1}{3}(C, D) + \frac{1}{3}(D, C) + \frac{1}{3}(C, C))$ is a correlated equilibrium, yielding payoffs $[3\frac{1}{3}, 3\frac{1}{3}]$ outside any convex combination of Nash equilibria.

To briefly see that this is a correlated equilibrium, consider the "row player" 1 (same works for player 2). If it is recommended to play C, its expected payoff is $\frac{1}{2} \cdot 4 + \frac{1}{2} \cdot 1 = \frac{5}{2}$ since, conditioned on $a_1 = C$, player 2 is recommended to play C and D with probability $\frac{1}{2}$ each. If player 1 switched to D, its expected payoff would still be $\frac{1}{2} \cdot 5 + \frac{1}{2} \cdot 0 = \frac{5}{2}$, making player 1 reluctant to switch. Similarly, if player 1 is recommended D, it knows that player 2 plays C (as (D, D) is never played in s^*), so its payoff is 5. Since this is the maximum payoff of the game, player 1 would not benefit by switching to C in this case. Thus, we indeed have a correlated equilibrium, where each player's payoff is $\frac{1}{3}(1 + 5 + 4) = 3\frac{1}{3}$, as claimed.

3 Implementing the Mediator

In this section we show how to remove the mediator using cryptographic means. We assume the existence of generic secure two-party protocols and show how to achieve our goal by using such protocols in the *game-theoretic* (rather than its designated cryptographic) setting. In other words, the players remain selfish and rational, even when running the cryptographic protocol. In Section 4 we give an efficient implementation for the types of cryptographic protocols that we need.

Extended Games. To remove the mediator, we assume that the players are (1) computationally bounded and (2) can communicate prior to playing the original game, which we believe are quite natural and minimalistic assumptions. To formally define the computational power of the players, we introduce an external security parameter into the game, and require that the strategies of both players can be computed in probabilistic polynomial time in the security parameter.[3]

To incorporate communication into the game, we consider an *extended game*, which is composed of three parts: first the players are given the security parameter and they freely exchange messages (i.e., execute any two-party protocol), then each player locally selects its move, and finally both players execute their move simultaneously.The final payoffs u'_i of the extended game are just the corresponding payoffs of the original game applied to the players' simultaneous moves at the last step.

The notions of a strategy and a strategy profile are straightforwardly generalized from those of the basic game, except that they are full-fledged probabilistic algorithms telling each player what to do *in each situation*. We now define the notion of a *computational Nash equilibrium* of the extended game, where the strategies of both players are restricted to probabilistic polynomial time (PPT). Also, since we are talking about a computational model, the definition must account for the fact that the players may break the underlying cryptographic scheme with negligible probability (e.g., by guessing the secret key), thus gaining some advantage in the game. In the definition and discussion below, we denote by $negl(k)$ some function that is negligible in k.

Definition 3. *A* computational Nash equilibrium *of an extended game G is an independent strategy profile (σ_1^*, σ_2^*), such that*

(a) both σ_1^*, σ_2^* are PPT computable; and
(b) for any other PPT computable strategies σ'_1, σ'_2, we have
$u_1(\sigma'_1, \sigma_2^*) \leq u_1(\sigma_1^*, \sigma_2^*) + negl(k)$ *and* $u_2(\sigma_1^*, \sigma'_2) \leq u_2(\sigma_1^*, \sigma_2^*) + negl(k)$.

We notice that the new "philosophy" for both players is still to maximize their expected payoff, except that the players will not change their strategy if their gain is negligible.

The idea of getting rid of the mediator is now very simple. Consider a correlated equilibrium $s(A_1 \times A_2)$ of the original game G. Recall that the job of the mediator is to sample a pair of actions (a_1, a_2) according to the distribution s, and to give a_i to player i. We can view the mediator as a trusted party who securely computes a probabilistic (polynomial-time) function s. Thus, to remove it we can have the two players execute a cryptographic protocol P that securely computes the function s. The strategy of each player would be to follow the protocol P, and then play the action a that it got from P.

Yet, several issues have to be addressed in order to make this idea work. First, the above description does not completely specify the strategies of the players. A full specification of a strategy must also indicate what a player should do if the other player *deviates* from its strategy (in our case, does not follow the protocol P). While cryptography does not address this question (beyond the guarantee that the other player is likely to detect the deviation and abort the protocol), it is

[3] Note that the parameters of the original game (like the payoff functions, the correlated equilibrium distribution, etc.) are all independent of the security parameter, and thus can always be computed "in constant time".

crucial to resolve it in our setting, since “the game must go on”: No matter what happens inside P, both players eventually have to take simultaneous actions, and receive the corresponding payoffs (which they wish to maximize). Hence we must explain how to implement a “punishment for deviation” within the game-theoretic framework.

Punishment for Deviations. We employ the standard game-theoretic solution, which is to punish the cheating player to his *minmax level*. This is the smallest payoff that one player can “force” the other player to have. Namely, the minmax level of player 2 is $\underline{v_2} = \min_{s_1} \max_{s_2} u_2(s_1, s_2)$. Similarly, minmax level of player 1 is $\underline{v_1} = \min_{s_2} \max_{s_1} u_1(s_1, s_2)$. To complete the description of our proposed equilibrium, we let each player punish the other player to its minmax level, if the other player deviates from P and is “caught”. Namely, if player 2 cheats, player 1 will play in the last stage of the game the strategy $\underline{s_1}$ achieving the minmax payoff $\underline{v_2}$ for player 2 and vice versa. Note that the instances where a player deviates from P but this is not detected falls under the negligible probability that the protocol will fail. Note also that in “interesting” games, the minmax payoff would be strictly smaller than the correlated equilibrium payoffs. Intuitively, in this case the only potentially profitable cheating strategy is an “honest but curious” behavior, where a player follows the prescribed protocol but tries nonetheless to learn additional information about the action of the other player. Any other cheating strategy would carry an overwhelming probability of “getting caught”, hence causing a real loss. Thus, we first observe the following simple fact:

Lemma 1. *Let $s^* = (s_1^*, s_2^*)$ be a correlated equilibrium. For any action a_1 of player 1 which occurs with non-zero probability in s^*, denote $\mu_1(a_1) = u_1(a_1, s_2^*|a_1)$. That is, $\mu(a_1)$ is the expected payoff of player 1 when its recommended action is a_1. Similarly, we define for player 2 $\mu_2(a_2) = u_2(s_1^*|a_2, a_2)$.*

Let $\underline{v_i}$ be the minmax payoff of player i, then for every a_1, a_2 that occur with non-zero probability in s^*, it holds that $\mu_i(a_i) \geq \underline{v_i}$.

Theorem 1 now follows almost immediately from Lemma 1 and the security of P. Intuitively, since (a) a cheating player that “gets caught” is going to lose by Lemma 1 and (b) the security of P implies that cheating is detected with very high probability, we get that the risk of getting caught out-weights the benefits of cheating, and players will not have an incentive to deviate from the protocol P. (A particular type of cheating in P is “early stopping”. Since the extended game must always result in players getting payoffs, early stopping is not an issue in game theory, since it will be punished by the minmax level as well.)

Somewhat more formally, let $v_1 = u_1(s_1^*, s_2^*)$, and consider that 1 is a cheating player who uses some arbitrary (but PPT computable) strategy s_1' (the analysis for player 2 is similar). Let the action taken by player 1 in the extended game be considered its output of the protocol. The output of player 2 is whatever is specified in its part of the protocol P, which is either an action (if the protocol runs to completion) or “abort” (if some “cheating” is detected).

According to standard definitions of secure protocols (e.g., the one by Canetti [9]), P is secure if the above output pair can be simulated in an “ideal model”. This "ideal model" is almost exactly the model of the trusted mediator, except that player 1 may choose to have the mediator abort before it recommends an action to player 2 (in which case the output of player 2 in the ideal model is also "abort"). The security of P implies that the output distribution in the execution of the protocol in the "real world" is indistinguishable from that of the "ideal world".

Consider now the function $\tilde{u}_1(\cdot, \cdot)$, which denotes the "payoff of player 1" in the extended game, given a certain output pair. That is, if the output is a pair of actions (a_1, a_2) than $\tilde{u}_1(a_1, a_2) = u_1(a_1, a_2)$, and if the output of the second player is "abort" then $\tilde{u}_1(a_1, \text{“abort”}) = u_1(a_1, \underline{a_2})$, where $\underline{a_2}$ is the minmax move for player 2. Note that in the real world, the function $\tilde{u}_1$ indeed represents the payoff of player 1 using strategy s'_1, but note also that this function is well defined even in the ideal world. Clearly, the expected value of $\tilde{u}_1$ in the real world is at most negligibly higher than in the ideal world. Otherwise, the output distributions in the two worlds could be distinguished with a non-negligible advantage by comparing the value of this function to some properly chosen threshold, contradicting the security of the protocol P.

Therefore, to prove Theorem 1 it is sufficient to show that the expected value of $\tilde{u}_1$ in the ideal world is at most v_1 (which is equal to the correlated equilibrium payoff of player 1 in the original game G). This is where we use Lemma 1: this lemma tells us that in the ideal world, no matter what action that is recommended to player 1, this player cannot increase the expected value of $\tilde{u}_1$ by aborting the mediator before it recommends an action to player 2. Hence, we can upper bound the expected value of $\tilde{u}_1$ in the ideal world by considering a strategy of player 1 that never aborts the mediator. Such strategy corresponds exactly to a strategy in the original game G (with the mediator), and so it cannot achieve expected payoff of more than v_1. This completes the proof.

Subgame Perfect Equilibrium. In looking at the computational Nash equilibrium we constructed, one may wonder why would a player want to carry out the "minmax punishment" when it catches the other player cheating (since this "punishment" may also hurt the "punishing player"). The answer is that the notion of Nash equilibrium only requires player's actions to be optimal *provided the other player follows its strategy.* Thus, it is acceptable to carry out the punishment even if this results in a loss for *both* players. We note that this oddity (known as an "empty threat" in the game-theoretic literature) is one of the reason the concept of Nash equilibrium is considered weak in certain situations. As a result, game theorists often consider a stricter version of a Nash equilibrium for extended games, called a *subgame perfect* equilibrium.

In the full version we show that Theorem 1 can be broadened to the case of the subgame perfect equilibrium. Generally stated, we prove that every "interesting" correlated-equilibrium payoff of the game G can be achieved by a subgame perfect equilibrium of an extended game G'.

4 The Correlated Element Selection Problem

In most common games, the joint strategy of the players is described by a short list of pairs $\{(\mathsf{move1}, \mathsf{move2})\}$, where the strategy is to choose at random one pair from this list, and have Player 1 play $\mathsf{move1}$ and Player 2 play $\mathsf{move2}$. (For example, in the game of chicken the list consists of three pairs $\{(D,C),(C,D),(C,C)\}$.)[4]

Hence, to obtain an efficient solution for such games, we need an efficient cryptographic protocol for the following problem: Two players, A and B, know a list of pairs $(a_1,b_1),\ldots,(a_n,b_n)$ (maybe with repetitions), and they need to jointly choose a random index i, and have player A learn only the value a_i and player B learn only the value b_i. We call this problem the *Correlated Element Selection* problem. In this section we describe our efficient solution for this problem. We start by presenting some notations and tools that we use (in particular, "blindable encryption schemes"). We then show a simple protocol that solves this problem in the special case where the two players are "honest but curious", and explain how to modify this protocol to handle the general case where the players can be malicious.

4.1 Notations and Tools

We denote by $[n]$ the set $\{1,2,\ldots n\}$. For a randomized algorithm A and an input x, we denote by $A(x)$ the output distribution of A on x, and by $A(x;r)$ we denote the output string when using the randomness r. If one of the inputs to A is considered a "key", then we write it as a subscript (e.g., $A_k(x)$). We use $pk, pk_1, pk_2, \ldots$ to denote public keys and $sk, sk_1, sk_2, \ldots$ to denote secret keys.

The main tool that we use in our protocol is *blindable encryption schemes.* Like all public-key encryption schemes, blindable encryption schemes include algorithms for key-generation, encryption and decryption. In addition they also have a "blinding" and "combining" algorithms. We denote these algorithms by *Gen*, *Enc*, *Dec*, *Blind*, and *Combine*, respectively. Below we formally define the blinding and combining functions. In this definition we assume that the message space M forms a group (which we denote as an additive group with identity 0).

Definition 4 (Blindable encryption). *A public-key encryption scheme $\mathcal{E}$ is* blindable *if there exist (PPT) algorithms Blind and Combine such that for every message m and every ciphertext $c \in Enc_{pk}(m)$:*

- *For any message m' (also referred to as the "blinding factor"), $Blind_{pk}(c,m')$ produces a* random *encryption of $m+m'$. Namely, the distribution $Blind_{pk}(c,m')$ should be equal to the distribution $Enc_{pk}(m+m')$.*

$$Enc_{pk}(m+m') \equiv Blind_{pk}(c,m') \qquad (1)$$

[4] Choosing from the list with distribution other than the uniform can be accommodated by having a list with repetitions, where a high-probability pair appears many times.

- *If r_1, r_2 are the random coins used by two successive "blindings", then for any two blinding factors m_1, m_2,*

$$Blind_{pk}(Blind_{pk}(c, m_1;\ r_1), m_2;\ r_2) = Blind_{pk}(c, m_1{+}m_2;\ Combine_{pk}(r_1, r_2)) \tag{2}$$

Thus, in a blindable encryption scheme anyone can "randomly translate" the encryption c of m into an encryption c' of $m + m'$, without knowledge of m or the secret key, and there is an efficient way of "combining" several blindings into one operation.

Both the ElGamal and the Goldwasser-Micali encryption schemes can be extended into blindable encryption schemes. We note that most of the components of our solution are independent of the specific underlying blindable encryption scheme, but there are some aspects that still have to be tailored to each scheme. (Specifically, proving that the key generation process was done correctly is handled differently for different schemes. See details in the full paper [13].)

4.2 A Protocol for the Honest-but-Curious Case

For the case of honest-but-curious players, one can present an "almost trivial" solution using any 1-out-of-n oblivious transfer protocol. However, in order to be able to derive an efficient protocol also for the general case, our starting point would be a somewhat different (but still very simple) protocol.

Let us recall the Correlated Element Selection problem. Two players share a public list of pairs $\{(a_i, b_i)\}_{i=1}^{n}$. For reasons that will soon become clear, we call the two players the "Preparer" (P) and the "Chooser" (C). The players wish to pick a random index i such that P only learns a_i and C only learns b_i. Figure 1 describes the Correlated Element Selection protocol for the honest-but-curious players. We employ a semantically secure blindable encryption scheme and for simplicity, we assume that the keys for this scheme were chosen by a trusted party ahead of time and given to P, and that the public key was also given to C.

At the beginning of the protocol, the Preparer randomly permutes the list, encrypts it element-wise and sends the resulting list to the Chooser. (Since the encryption is semantically secure, the Chooser "cannot extract any useful information" about the permutation π.) The Chooser picks a random pair of ciphertexts (c_ℓ, d_ℓ) from the permuted list (so the final output pair will be the decryption of these ciphertexts). It then blinds c_ℓ with 0 (i.e. makes a random encryption of the same plaintext), blinds d_ℓ with a random blinding factor β, and sends the resulting pair of ciphertexts (e, f) back to the Preparer. Decryption of e gives the Preparer its element a (and nothing more, since e is a *random* encryption of a after the blinding with 0), while the decryption $\tilde{b}$ of f does not convey the value of the actual encrypted message since it was blinded with a random blinding factor. The Preparer sends $\tilde{b}$ to the Chooser, who recovers his element b by subtracting the blinding factor β.

It is easy to show that if both players follow the protocol then their output is indeed a random pair (a_i, b_i) from the known list. Moreover, at the end of the

Protocol CES-1

Common inputs: List of pairs $\{(a_i, b_i)\}_{i=1}^{n}$, public key pk.
Preparer knows: secret key sk.

P : **1. Permute and Encrypt**.
Pick a random permutation π over $[n]$.
Let $(c_i, d_i) = (Enc_{pk}(a_{\pi(i)}),\ Enc_{pk}(b_{\pi(i)}))$, for all $i \in [n]$.
Send the list $\{(c_i, d_i)\}_{i=1}^{n}$ to C.

C : **2. Choose and Blind**.
Pick a random index $\ell \in [n]$, and a random blinding factor β.
Let $(e, f) = (Blind_{pk}(c_\ell, 0),\ Blind_{pk}(d_\ell, \beta))$.
Send (e, f) to P.

P : **3. Decrypt and Output**.
Set $a = Dec_{sk}(e)$, $\tilde{b} = Dec_{sk}(f)$. Output a.
Send $\tilde{b}$ to C.

C : **4. Unblind and Output**.
Set $b = \tilde{b} - \beta$. Output b.

Fig. 1. Protocol for Correlated Element Selection in the honest-but-curious model.

protocol the Preparer has no information about b other than what's implied by its own output a, and the Chooser gets "computationally no information" about a other than what's implied by b. Hence we have:

Theorem 2. *Protocol* CES-1 *securely computes the (randomized) function of the Correlated Element Selection problem in the honest-but-curious model.*

Proof omitted.

4.3 Dealing with Dishonest Players

Generic transformation. Following the common practice in the design of secure protocols, one can modify the above protocol to deal with dishonest players by adding appropriate zero-knowledge proofs. That is, after each flow of the original protocol, the corresponding player proves in zero knowledge that it indeed followed its prescribed protocol: After Step 1, the Preparer proves that it knows the permutation π that was used to permute the list. After Step 2 the Chooser proves that it knows the index ℓ and the blinding factor that was used to produce the pair (e, f). Finally, after Step 3 the Preparer proves that the plaintext $\tilde{b}$ is indeed the decryption of the ciphertext f. Given these zero-knowledge proofs, one can appeal to general theorems about secure two-party protocols, and prove that the resulting protocol is secure in the general case of potentially malicious players.

We note that the zero-knowledge proofs that are involved in this protocol can be made very efficient, so even this “generic” protocol is quite efficient (these are essentially the same proofs that are used for mix-networks in [1], see description in the full paper). However, a closer look reveals that one does not need all the power of the generic transformation, and the protocol can be optimized in several ways. Some of the optimizations are detailed below, while protocols for the zero-knowledge proofs and issues of key generation can be found in the full paper [13]. The resulting protocol CES-2 is described in Figure 2.

Theorem 3. *Protocol* CES-2 *securely computes the (randomized) function of the Correlated Element Selection problem.*

Proof omitted.

Proof of proper decryption. To withstand malicious players, the Preparer P must “prove” that the element $\tilde{b}$ that it send in Step 3 of CES-1 is a proper decryption of the ciphertext f. However, this can be done in a straightforward manner without requiring zero-knowledge proofs. Indeed, the Preparer can reveal additional information (such as the randomness used in the encryption of f), as long as this extra information does not compromise the semantic security of the ciphertext e. The problem is that P may not be able to compute the randomness of the blinded value f (for example, in ElGamal encryption this would require computation of discrete log). Hence, we need to devise a different method to enable the proof.

The proof will go as follows: for each $i \in [n]$, the Preparer sends the element $b_{\pi(i)}$ and corresponding random string that was used to obtain ciphertexts d_i in the first step. The Chooser can then check that the element d_ℓ that it chose in Step 2 was encrypted correctly, and learn the corresponding plaintext.

Clearly, in this protocol the Chooser gets more information than just the decryption of f (specifically, it gets the decryption of all the d_i’s). However, this does not affect the security of the protocol, as the Chooser now sees a decryption of a permutation of a list that he knew at the onset of the protocol. This permutation of the all b_i’s does not give any information about the output of the Preparer, other than what is implied by its output b. In particular, notice that if b appears more than once in the list, then the Chooser does not know which of these occurrences was encrypted by d_ℓ.

Next, we observe that after the above change there is no need for the Chooser to send f to the Preparer; it is sufficient if C sends only e in Step 2, since it can compute the decryption of d_ℓ by itself.

A weaker condition in the second proof-of-knowledge. Finally, we observe that since the security of the Chooser relies on an information-theoretic argument, the second proof-of-knowledge (in which the Chooser proves that it knows the index ℓ) does not have to be fully zero-knowledge. In fact, tracing through the proof of security, one can verify that it is sufficient for this proof to be *witness independent* in the sense of Feige and Shamir [16].

Protocol CES-2

Common inputs: List of pairs $\{(a_i, b_i)\}_{i=1}^n$, public key pk.
Preparer knows: secret key sk.

P : **1. Permute and Encrypt**.
Pick a random permutation π over $[n]$, and random strings $\{(r_i, s_i)\}_{i=1}^n$.
Let $(c_i, d_i) = (Enc_{pk}(a_{\pi(i)}; r_{\pi(i)}),\ Enc_{pk}(b_{\pi(i)}; s_{\pi(i)}))$, for all $i \in [n]$.
Send $\{(c_i, d_i)\}_{i=1}^n$ to C.

Sub-protocol Π_1: P proves in zero-knowledge that it knows the randomness $\{(r_i, s_i)\}_{i=1}^n$ and permutation π that were used to obtain the list $\{(c_i, d_i)\}_{i=1}^n$.

C : **2. Choose and Blind**.
Pick a random index $\ell \in [n]$.
Send to P the ciphertext $e = Blind_{pk}(c_\ell, 0)$.

Sub-protocol Π_2: C proves in a witness-independent manner that it knows the randomness and index ℓ that were used to obtain e.

P : **3. Decrypt and Output**.
Set $a = Dec_{sk}(e)$. Output a.
Send to C the list of pairs $\{(b_{\pi(i)}, s_{\pi(i)})\}_{i=1}^n$ (in this order).

C : **4. Verify and Output**.
Denote by (b, s) the ℓ'th entry in this lists (i.e., $(b, s) = (b_{\pi(\ell)}, s_{\pi(\ell)})$).
If $d_\ell = Enc_{pk}(b; s)$ then output b.

Fig. 2. Protocol for Correlated Element Selection.

Blinding by Zero. Notice that for the modified protocol we did not use the full power of blindable encryption, since we only used "blindings" by zero. Namely, all that was used in these protocols is that we can transform any ciphertext c into a *random* encryption of the same plaintext. (The zero-knowledge proofs also use only "blindings" by zero.) This is exactly the "random self-reducibility" property used by Sander et al. [33].

Efficiency. We note that all the protocols that are involved are quite simple. In terms of number of communication flows, the key generation step and Step 1 take at most five flows each, using techniques which appear in Appendix A. Step 2 takes three flows and Step 3 consists of just one flow. Moreover, these flows can be piggybacked on each other. Hence, we can implement the protocol with only five flows of communication, which is equal to the five steps which are required by a single proof. In terms of number of operations, the complexity of the protocol is dominated by the complexity of the proofs in Steps 1 and 2. The proof in Step 1 requires nk blinding operations (for a list of size n and security parameter k), and the proof of Step 2 can be optimized to about $nk/2$ blinding operations on the average. Hence, the whole protocol has about $\frac{3}{2}nk$ blinding operations.[5]

5 Epilogue: Cryptography and Game Theory

The most interesting aspect of our work is the synergy achieved between cryptographic solutions and the game-theory world. Notice that by implementing our cryptographic solution in the game-theory setting, we gain on the game-theory front (by eliminating the need for a mediator), but we also gain on the cryptography front (for example, in that we eliminate the problem of early stopping). In principle, it may be possible to make stronger use of the game theory setting to achieve improved solutions. For example, maybe it is possible to prove that in the context of certain games, a player does not have an incentive to deviate from its protocol, and so in this context there is no point in asking this player to prove that it behaves honestly (so we can eliminate some zero-knowledge proofs that would otherwise be required).

More generally, it may be the case that working in a model in which "we know what the players are up to" can simplify the design of secure protocols. It is a very interesting open problem to find interesting examples that would demonstrate such phenomena.

We conclude with the table that shows some parallels between Cryptography and Game Theory that we discussed.

Issue	Cryptography	Game Theory
Incentive	None	Payoff
Players	Totally Honest/Malicious	Always Rational
Punishing Cheaters	Outside Model	Central Part
Solution Concept	Secure Protocol	Equilibrium
Early Stopping	Problem	Not an Issue

References

1. M. Abe. Universally Verifiable Mix-net with Verification Work Independent on the number of Mix-centers. In *Proceedings of EUROCRYPT '98*, pp. 437-447, 1998.
2. R. Aumann. Subjectivity and Correlation in Randomized Strategies. In *Journal of Mathematical Economics*, 1, pp. 67-95, 1974

[5] We note that the protocol includes just a single decryption operation, in Step 3. In schemes where encryption is much more efficient than decryption – such as the Goldwasser-Micali encryption – this may have a significant impact on the performance of the protocol.

3. I. Barany. Fair distribution protocols or how the players replace fortune. *Mathematics of Operations Research*, 17(2):327–340, May 1992.
4. M. Bellare, R. Impagliazzo, and M. Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th Annual Symposium on Foundations of Computer Science*, pages 374–383. IEEE, 1997.
5. J. Benaloh. Dense Probabilistic Encryption. In *Proc. of the Workshop on Selected Areas in Cryptography*, pp. 120-128, 1994.
6. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988.
7. M. Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *CRYPTO '81*. ECE Report 82-04, ECE Dept., UCSB, 1982.
8. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *JCSS*, 37(2):156–189, 1988.
9. R. Canetti, Security and Composition of Multi-parti Cryptographic Protocols. *Journal of Cryptology*, 13(1):143–202.
10. D. Chaum. Blind signatures for untraceable payment. In *Advances in Cryptology – CRYPTO '82*, pages 199–203. Plenum Press, 1982.
11. D. Chaum, C. Crépeau, and E. Damgård. Multiparty unconditionally secure protocols. In *Advances in Cryptology – CRYPTO '87*, volume 293 of *99 Lecture Notes in Computer Science*, pages 462–462. Springer-Verlag, 1988.
12. R. Cramer, I. Damgard, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. Proceedings of *PKC 2000* January 2000, Melbourne, Australia.
13. Y. Dodis and S. Halevi and T. Rabin. Cryptographic Solutions to a Game Theoretic Problem. `http://www.research.ibm.com/security/DHR00.ps`.
14. C. Dwork, M. Naor, and A. Sahai. Concurrent zero knowledge. In *Proceedings of the 30th Annual ACM STOC* , pages 409–418. ACM Press, 1998.
15. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO '84*, *LNCS 196*, pages 10–18. Springer-Verlag, 1985.
16. U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM STOC* , pages 416–426. ACM Press, 1990.
17. M. Fischer, R. Wright. An Application of Game-Theoretic Techniques to Cryptography. In *Advances in Computational Complexity Theory*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 13, pp. 99–118, 1993.
18. F. Forges. Can sunspots repalce the mediator? In J. of Math. Economics, 17:347–368, 1988.
19. F. Forges. Universal Mechanisms, In Econometrica, 58:1341–1364, 1990.
20. D. Fudenberg, J. Tirole. Game Theory. MIT Press, 1992.
21. J. Garay, R. Gennaro, C. Jutla, and T. Rabin. Secure distributed storage and retrieval. In *Proc. 11th International Workshop on Distributed Algorithms (WDAG '97)*, *LNCS 1320*, pages 275–289. Springer-Verlag, 1997.
22. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 218–229, 1987.
23. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.
24. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
25. M. Jakobsson. A Practical Mix. In *Proceedings of EUROCRYPT '98*, pp. 448–461, 1998.

26. J. Kilian. (More) Completeness Theorems for Secure Two-Party Computation In *Proc. of STOC*, 2000.
27. E. Lehrer and S. Sorin. One-shot public mediated talk. Discussion Paper 1108, Northwestern University, 1994.
28. P. MacKenzie. Efficient ZK Proofs of Knowledge. Unpublished manuscript, 1998.
29. G. Mailath, L. Samuelson and A. Shaked. Correlated Equilibria and Local Interaction In *Economic Theory*, 9, pp. 551-556, 1997.
30. R. Myerson. Communication, correlated equilibria and incentive compatibility. In *Handbook of Game Theory*, Vol. II, Elsevier, Amsterdam, pp. 827-847, 1994.
31. J.F. Nash. Non-Cooperative Games. *Annals of Mathematics,* 54 pages 286–295.
32. M. Osborne, A. Rubinstein. A Course in Game Theory. The MIT Press, 1994.
33. T. Sander, A. Young, and M. Yung. Non-interactive CryptoComputing for NC1. In *40th Annual Symposium on Foundations of Computer Science*, pages 554–567. IEEE, 1999.
34. A. C. Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, Nov. 1982.

A Reducing the Error in a Zero-knowledge Proof-of-knowledge

Below we describe a known transformation from any 3-round, constant-error zero-knowledge proof-of-knowledge into a 5-round, negligible error zero-knowledge proof-of-knowledge, that uses trapdoor commitment schemes. We were not able to trace the origin of this transformation, although related ideas and techniques can be found in [14, 28, 12].

Assume that you have some 3-round, constant-error zero-knowledge proof-of-knowledge protocol, and consider the 3-round protocol that you get by running the constant-error protocol many times in parallel. Denote the first prover message in the resulting protocol by α, the verifier message by β, and the last prover message by γ. Note that since the original protocol was 3-round, then parallel repetition reduces the error exponentially (see proof in [4]). However, this protocol is no longer zero-knowledge.

To get a zero-knowledge protocol, we use a trapdoor (or *Chameleon*) commitment schemes [8]. Roughly, this is a commitment scheme which is computationally binding and unconditionally secret, with the extra property that there exists a trapdoor information, knowledge of which enables one to open a commitment in any way it wants.

In the zero-knowledge protocol, the prover sends to the verifier in the first round the public-key of the trapdoor commitment scheme. The verifier then commits to β, the prover sends α, the verifier opens the commitment to β, and the prover sends γ and also *the trapdoor for the commitment*. The zero-knowledge simulator follows the one for the standard 4-round protocol. The knowledge extractor, on the other hand, first runs one instance of the proof to get the trapdoor, and then it can effectively ignore the commitment in the second round, so you can use the extractor of the original 3-round protocol.