Applying Cryptography to achieve Optimality in a Game-Theoretic problem

Thesis submitted in partial fulfilment of the requirement for the degree of

Bachelor of Science

In

Computer Science

Under the Supervision of

Abu Mohammad Hammad Ali

And

Co-Supervision of

Dr. Wahid Abdallah

By

Adnan Reza (14141012),

Akib Mahmud (14341014),

Shaila Sabrin (14141013)

School of Engineering & Computer Science

Department of Computer Science & Engineering

BRAC University

Declaration

We hereby declare that this thesis is based on results obtained from our own work. Due acknowledgement has been made in the text to all other material used. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor:

_____

Abu Mohammad Hammad Ali

Department of Computer Science & Engineering,
BRAC University

Supervisor

Signature of Authors:

_____

Adnan Reza, 14141012

_____

Akib Mahmud, 14341014

_____

Shaila Sabrin, 14141013

Acknowledgements

We would like to start by thanking our thesis supervisor Mr. Hammad Ali for allowing us to work on this thesis under his supervision and for his continuous support and guidance. He introduced us to the exciting and challenging field of theoretical computer science during our third year and has been a continuous source of inspiration over the past four years.

We would also like to thank our co-supervisor Dr. Wahid Abdullah for extending every possible help when asked for and giving his valuable time to discuss ideas and help with core game-theoretic concepts.

We are also grateful to Coursera and Udacity for the free online resources on Cryptography and Game Theory.

Abstract

We aim to use cryptography to solve a game-theoretic problem which is prevalent in the area of two party strategic games. The standard game-theoretic solutions concept for such games is that of a Nash equilibrium: a pair of "self-enforcing" strategies which makes 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 i.e. a "mediator" assists in choosing their moves (correlated equilibria), than when each play has to choose its move on its own (Nash Equilibria). It is natural to ask whether there exists a mechanism (cryptographic protocol) 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 by extending the original game by adding an initial step in which the two players communicate, and then proceed to execute the game as usual. By incorporating our cryptographic protocol into a game-theoretic setting, we hope to highlight some interesting parallels between cryptographic protocols and two-party games. An interesting aspect of our work is the synergy achieved between cryptographic algorithms and game-theoretic problems: By implementing the cryptographic protocol in the game theoretic problem, we gain in the game theory front by eliminating the need for the mediator; we also gain on the cryptography front: for instance, we eliminate the problem of early stopping.

Table of Contents

1. Introduction

Game theory is the study of multi-agent decision problems. While Game Theory was initially a mathematical tool used in the field of microeconomics to model strategic decision making among rational agents (particularly firms and humans), today it is extensively used in a diverse range of fields including computer science, political science, psychology and biology. It applies to a wide range of behavioural relations, and has developed into an umbrella term for the logical side of decision science, including both humans and non-humans (e.g. computers, insects, animals).

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about designing protocols that overcome the influence of adversaries. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means.

1.1 Motivation

The research areas of Game Theory and Cryptography are both extensively studied fields with many problems and solutions. Yet, the overlap between them is surprisingly small. It is seldom the case that tools from one area are borrowed to address problems in the other. In this thesis, we demonstrate the benefits which arise in such a combined and interdisciplinary setting. Specifically, we show how cryptographic tools can be used to address a natural problem in the Game Theory world. We make use of the fact that both fields are concerned with the study of interactions among mutually distrusting parties. We hope that this work will encourage greater synergy between these classical fields.

1.2 The inherent inefficiency of the Nash Equilibrium

The standard game-theoretic solutions concept for *n*-party strategic games is that of a Nash equilibrium: a pair of "self-enforcing" strategies which makes each player's strategy an optimal response to the other player's strategy [1]. It is known that for many games the expected equilibrium payoffs can be much higher when a trusted third party i.e. a "mediator" assists in choosing their moves (correlated equilibria) [2], than when each player has to choose its move on its own (Nash Equilibria). It is natural to ask whether there exists a mechanism (cryptographic protocol) that eliminates the need for the mediator yet allows the players to maintain the high payoffs offered by mediator-assisted strategies.

1.3 Addressing this inefficiency: Removing the Mediator

We answer this question by extending the game by adding an initial step in which the two players communicate, and then proceed to execute the game as usual.

As the game was intended for two players, it is natural to ask if correlated equilibria [2] can be implemented without the mediator. In the language of cryptography, we ask if we can design a two party game to eliminate the third player (the mediator) from the original game [5]. 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 ([7, 8, 14]). We show that this positive answer carries over also to the Game Theory model. Specifically, we consider an *extended game*, in which the players first exchange messages, and then they choose their actions and execute them simultaneously. The payoffs are still computed as a function of their actions or moves, according to the same payoff function as in the original game. Also, we define a *computational Nash equilibrium* as one where the strategies of both players are restricted to probabilistic polynomial time [5].

Thus, the mediator can be avoided if the players are computationally bounded and can communicate prior to the game. We stress that although this point of view is quite natural from a cryptography point of view, the models of Game Theory and Cryptography are different, thus applying 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 [5]. In the game-theoretic setting, on the other hand, *both players are selfish and rational*: players are expected to deviate from the protocol if the deviation results in a higher utility, and we assume players follow their protocol otherwise. Also, it is important to realize that in this setting we cannot use cryptography to "enforce" honest behavior [5]. The only thing that the players are able to do is to choose their moves and execute them at the end of the game. Therefore, even the most elaborate protocol would be ineffective if a "cheating agent" can simply ignore the fact that it was "caught cheating" during the protocol, and nonetheless choose a move that maximizes its profit. We elaborate more on this issue in Section 3.

1.4. Literature Review

Realizing the advantages of removing the mediator, various papers in the Game Theory community have been published to try and achieve this goal. Barany [23] substitutes the trusted mediator with four, potentially untrusted players. These players, two of which were the actual players who needed to play the game in a distributive manner and privately computed the moves for the two active players. This protocol works in an information-theoretic setting, which explains the need for four players [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. [5, 17]) can also be used, even though the solution of [3] is simpler and more efficient. The work of Lehrer and Sorin [10] describes protocols that "reduce" the role of the mediator: the mediator in this protocol computes some function on values which the players chose.

2. Background in Game Theory

2.1 Two Player Strategic Games

The game-theoretic problem that we address here belongs to the general area of *two player strategic games*, a widely studied area in the game-theory community [1, 13]. 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 execute 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 method for choosing its move. The fundamental assumption of game theory is that each player is *selfish and rational* [13], 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 of achieving an equilibrium was introduced in the classical work of John Nash [1]. 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.)

2.2 Nash Equilibrium

Informally, a set of strategies is a Nash equilibrium if no player can do better by unilaterally changing his or her strategy [1, 13]. To see what this means, imagine that each player is told the strategies of the others. Suppose then that each player asks himself or herself: "Knowing the strategies of the other players, and treating the strategies of the other players as set in stone, can I benefit by changing my strategy?"

If any player would answer "Yes", then that set of strategies is not a Nash equilibrium. But if every player prefers not to switch (or is indifferent between switching and not) then the set of strategies is a Nash equilibrium. Thus, each strategy in a Nash equilibrium is a *best response* to all other strategies in that equilibrium [13].

The Nash equilibrium may sometimes appear non-rational in a third-person perspective. This is because it may happen that a Nash equilibrium is not Pareto optimal [5].

The Nash equilibrium may also have non-rational consequences in sequential games because players may "threaten" each other with non-rational moves [5]. For such games the subgame perfect Nash equilibrium may be more meaningful as a tool of analysis.

Formal definition [13]

Let (S, f) be a game with n players, where S_i is the strategy set for player i , $S = S_1 \times S_2 \times \cdots \times S_n$ is the set of strategy profiles and $f = (f_1(x), \ldots, f_n(x))$ is the payoff function for $x \in S$. Let x_i be a strategy profile of player i and x_{-i} be a strategy profile of all players except for player i. When each player $i \in \{1, \ldots, n\}$ chooses strategy x_i resulting in strategy profile $x = (x_1, \ldots, x_n)$ then player i obtains payoff $f_i(x)$. Note that the payoff depends on the strategy profile chosen, i.e., on the strategy chosen by player i as well as the strategies chosen by all the other players. A strategy profile $x^* \in S$ is a Nash equilibrium if no unilateral deviation in strategy by any single player is profitable for that player, that is:

$$\forall i, x_i \in S_i : f_i(x_i^*, x_{-i}^*) \geq f_i(x_i, x_{-i}^*).$$

When the inequality above holds strictly (with > instead of ≥) for all players and all feasible alternative strategies, then the equilibrium is classified as a *strict Nash equilibrium*. If instead, for some player, there is exact equality between x_i^* and some other strategy in the set S, then the equilibrium is classified as a *weak Nash equilibrium*.

A game can have either a pure-strategy or a mixed Nash Equilibrium. (In the latter a pure strategy is chosen stochastically with a fixed probability).

2.3 Correlated Equilibrium

While the Nash Equilibrium is a simple and intuitive means of achieving an equilibrium, it is not necessarily the most optimal or efficient in terms of overall utility or social welfare. 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*, the model of the game is modified and a "trusted third party", called a *mediator* is introduced. This mediator chooses the pair of moves according to the right distributions and privately tells each player what its designated move is. Since the strategies are correlated, the move of one player typically carries some information 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.

Informal definition

In game theory, a **correlated equilibrium** is a solution concept that is more general than the well-known Nash equilibrium. It was first discussed by mathematician Robert Aumann (1974). The idea is that each player chooses his/her action according to his/her observation of the value of the same public signal. A strategy assigns an action to every possible observation a player can make. If no player would want to deviate from the recommended strategy (assuming the others don't deviate), the distribution is called a correlated equilibrium.

Formal definition [13]

An N-player strategic game (N, A_i, u_i) is characterized by an action set A_i and utility function u_i for each player i. When player i chooses strategy $a_i \in A_i$ and the remaining players choose a strategy profile described by the $N-1$-tuple a_{-i}, then player i's utility is $u_i(a_i, a_{-i})$.

A *strategy modification* for player i is a function $\phi: A_i \to A_i$. That is, ϕ tells player i to modify his behavior by playing action $\phi(a_i)$ when instructed to play a_i.

Let (Ω, π) be a countable probability space. For each player i, let P_i be his information partition, q_i be i's posterior and let $s_i: \Omega \to A_i$, assigning the same value to states in the same cell of i's information partition. Then $((\Omega, \pi), P_i)$ is a correlated equilibrium of the strategic game (N, A_i, u_i) if for every player i and for every strategy modification ϕ:

$$\sum_{\omega \in \Omega} q_i(\omega) u_i(s_i, s_{-i}) \geq \sum_{\omega \in \Omega} q_i(\omega) u_i(\phi(s_i), s_{-i})$$

In other words, $((\Omega, \pi), P_i)$ is a correlated equilibrium if no player can improve his or her expected utility via a strategy modification.

2.4 A Worked Example: Game of "Chicken"

Let us consider a simple *2x2* game, the so called game of "Chicken" shown in the payoff matrix below:

		Player 2	
		Chicken Out (C)	**Dare (D)**
Player 1	**Chicken Out (C)**	**4,4**	**1,5**
	Dare (D)	**5,1**	**0, 0**

Figure 2.41:
Pure Strategy Nash Equilibrium
(D, C) and (C, D) are the two pure strategy Nash equilibria.

Here, each player can either "dare" (D) or "chicken out" (C).

- If both players "Dare", they collide and receive payoffs (0,0)
- If player 1 "Dares" & player 2 "Chickens out", then player 1 receives a payoff of 5 and player 2 receives only 1.
- If player 1 "Chickens Out" & player 2 "Dares", then player 1 receives a payoff of only 1 and player 2 receives 5.
- If players 1 and 2 both "Chicken Out". Then each receives a payoff of 4 and this is clearly the most socially optimal outcome!

While the wisest pair of actions is (C, C), this is not a Nash Equilibrium, since both players are willing to deviate to *D*. The game is easily seen to have three Nash Equilibria:

i. $s^1 = (D,C)$ with payoff *[5, 1]*

ii. $s^2 = (C,D)$ with payoff *[1, 5]*

iii. $s^3 = (0.5\ D + 0.5\ C,\ 0.5\ D + 0.5\ C)$ with payoff *[2.5, 2.5]*

We see that the first two strategies s^1 and s^2 are **pure strategy Nash Equilibria** and both are "unfair" as one player gets a significantly smaller payoff relative to the other (*1 vs 5*).

The third strategy $s^{3,}$ which is a **mixed strategy Nash equilibrium** has small payoffs, since the mutually desirable outcome (D, D) occurs with non-zero probability in the product distribution.

	Chicken Out (C)	**Dare (D)**
Chicken Out (C)	**1/4**	**1/4**
Dare (D)	**1/4**	**1/4**

Figure 2.42:
Mixed Nash Equilibrium
Each strategy (pair of actions) has a probability of 0.25 of being played.

On the other hand, the profile $s^c = (1/3^* (C, D) + 1/3^* (D, C) + 1/3 ^*(C, C))$ is a **correlated equilibrium**, yielding payoffs *(3.33, 3.33)* which is better than any convex combination of Nash equilibria.

	Chicken Out (C)	**Dare (D)**
Chicken Out (C)	**1/3**	**1/3**
Dare (D)	**1/3**	**0**

Figure 2.43:
Correlated Equilibrium
Strategies (C, C), (C, D) and (D, C) each has a 1/3 probability of being played. (D, D) is never played in $s^{c.}$

To briefly see it, let us consider the "row player", *player 1* (Refer to *figure 2.41* and *2.43*). If it is recommended to play C its expected payoff is (0.5 * 4 + 0.5 * 1) = 2.5 since conditioned on $a_1 = C$, *player 2* is recommended to play C and D with probability 0.5 each. If player 1 switched to D, its expected payoff is (0.5 * 5 + 0.5 * 0) = 2.5, 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 $s^{c)}$, 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 $1/3(1 + 5 + 4) = 3.33$.

We can clearly see how the inefficiency of Nash Equilibrium (both pure and mixed) can be overcome by the help of a third party that mediates and recommends actions to each player (correlated equilibrium). In the following two sections, we discuss how we achieve the same high payoffs offered by the correlated equilibrium after replacing the mediator with a cryptographic protocol.

3. Addressing the Inefficiency of Nash Equilibrium

3.1 Eliminating the Mediator

As mentioned earlier, we can achieve the same high payoffs offered by correlated equilibria (mediator assisted strategies) even after eliminating the mediator. We have to modify the game setting slightly in order to achieve this goal. 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 [5]. In other words, the players remain *selfish* and *rational*, even when running the cryptographic protocol.

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 [5].

To incorporate communication into the game, we consider an *extended game*, which is composed of three parts:

i. Initially, the players are given the security parameter and they freely exchange messages (i.e., execute any two-party protocol).
ii. Then each player locally selects its move.
iii. 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 players' simultaneous moves at the last step. The notions of a strategy and a strategy profile are trivially generalized from those of the basic game.

Similarly to the idea of a Nash equilibrium, we define a *computational Nash equilibrium* of the extended game, where the strategies of both players are restricted to probabilistic polynomial time. 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.

Probabilistic Polynomial Time

In complexity theory, **PP** is the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than 1/2 for all instances. The abbreviation **PP** refers to probabilistic polynomial time. The complexity class was defined by Gill in 1977 [24].

If a decision problem is in **PP**, then there is an algorithm for it that is allowed to flip coins and make random decisions. It is guaranteed to run in polynomial time.

Computational Nash Equilibrium

A computational Nash equilibrium [5] of an extended game *G* is an independent strategy profile (s_1^*, s_2^*) such that

(a) Both s_1^*, s_2^* *are PPT computable*

(b) For any other PPT computable strategies s_1', s_2' *there exists a negligible function* μ *such that on security parameter* k*, we have:*

$$u_1(s_1', s_2^*) <= u_1(s_1^*, s_2^*) + \mu(k) \quad \& \quad u_2(s_1^*, s_2') <= u_2(s_1^*, s_2^*) + \mu(k)$$

The idea of getting rid of the mediator is now very simple. Consider a Correlated equilibrium 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 recommend 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, and then play the action *a* that it got from the cryptographic protocol *P*.

Some Caveats

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. While cryptography does not address this question, it is crucial to resolve it in our setting, since "the game must go on": No matter what happens inside protocol P, both players eventually have to take simultaneous actions and receive the corresponding payoffs (which they wish to maximize). Therefore, we must explain how to implement a "punishment for deviation" within the game-theoretic framework.

3.2 Punishment for Deviations

We employ the standard game-theoretic solution, which is to punish the cheating player to his *minimax level*. This is the smallest payoff that one player can "force" the other player to have.

The minimax level of player 2 is: $v_2 = \min_{s1} \max_{s2} u_2 (s_1, s_2)$.

The minimax level of player 1 is $v_1 = \min_{s2} \max_{s1} u_1 (s_1, s_2)$.

To complete the description of our proposed equilibrium, we let each player punish the other player to its minimax level, if the other player deviates from its recommended action. Basically, if player 2 cheats, player 1 will play in the last stage of the game the strategy s_1 achieving the minimax payoff v_2 for player 2and vice versa.

Finally, if no player has cheated in P, the privacy of P implies that we achieved exactly the same effects as with the mediator: each player only learns its move, does not learn anything about the other player's move except for what is implied by its move (and possibly except for a negligible additional advantage). Since s is a Correlated equilibrium, both players will indeed take the action they outputted in P.

Why would a player want to carry out a "minimax punishment" strategy?

One question that may come to mind is why would a player want to carry out a "minimax punishment" when it catches the other player cheating. This is because this "punishment" may also hurt the "punishing player". However, the notion of Nash equilibrium only requires player's actions to be optimal *provided the other player follows its strategy* [13]. Thus, it is acceptable to carry out the punishment even if this results in a loss for *both* players. The cheating player should have been rational and should not have cheated in the first place.

4. Applying the Cryptographic Protocol

4.1 Background in Cryptography

In cryptography, **plaintext** is information a sender wishes to transmit to a receiver.

Encryption is the process of encoding messages or information in such a way that only authorized parties can read it**. Decryption** is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext **ciphertext** is the result of encryption performed on plaintext using an algorithm, called a cipher.

A **key** is a piece of information (a parameter) that determines the functional output of a cryptographic algorithm or cipher. Without a key, the algorithm would produce no useful result.

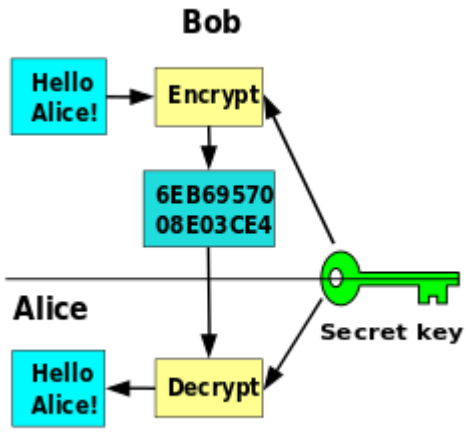

Figure 4.11:
Symmetric-Key Cryptography
A single key is used for both encryption and decryption.

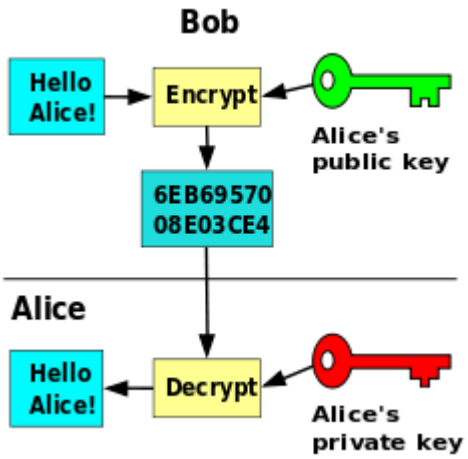

Figure 4.12:
Public-Key Cryptography
Different keys are used for both encryption and decryption.

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both

parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption

Public-key cryptography, also known as **asymmetric cryptography**, is a class of cryptographic algorithms which requires two separate keys, one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt ciphertext or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both. The strength lies in the fact that it is "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do *not* require a secure initial exchange of one (or more) secret keys between the parties.

A public key encryption scheme is **blindable** [5] if anyone can "randomly translate" the encryption of m into an encryption of $m + m'$, without knowledge of m or the secret key, and there is an efficient way of "combining" several blindings into one operation.

A **zero-knowledge proof** or **zero-knowledge protocol** is a method by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true.

4.2 The Cryptographic Protocol incorporated in the extended game

Now that we have eliminated the mediator, we modify the existing game so that the players can run the cryptographic protocol. The cryptographic protocol is based on the work done on the Correlated Element Selection Problem [5].

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-hiding 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.

Figure 4:
Cryptographic Protocol
The algorithm is based on work done on the Correlated Element Selection Problem.

4.3 How does the Protocol Work?

Step 0: Initially, the Preparer chooses the keys for the blindable encryption scheme, sends the public key to the Chooser and proves in zero-knowledge that the encryption is committing and has the blinding property. As we said above, this proof must be tailored to the particular encryption scheme that is used. Also, this step can be carried out only once, and the resulting keys can be used for many instances of the protocol.

Step 1: The Preparer encrypts the known list $\{(a_i, b_i)\}$ *from* $i = 1$ *to* n in some "canonical" manner, blinds with zero the list of ciphertexts, and permutes it with a random permutation π. It sends the resulting lists $\{(c_i, d_i)\}$ *from* $i = 1$ *to* n to the Chooser, and uses the protocol to prove in zero-knowledge that it knows the permutation that was used [5].

Step 2: The Chooser blinds with zeros the list of c_i's, and re-permutes it with a random permutation p. It sends the resulting list $\{e_i\}$ *from* $i = 1$ *to* n the Prover, and again uses the protocol to prove that it knows the permutation that was used. Here we can optimize the proof somewhat, since we later only use e_i and also because the proof only needs to be witness hiding.

Step 3: The Preparer decrypts the first ciphertext e_1 e1, and outputs the corresponding plaintext a. It also sends to the Chooser the list of the b_i's permuted according to π together with the randomness that was used to blind their "canonical encryption" to get the d_i's in Step 1.

Step 4: The Chooser C sets $l = p^{-1}(1)$ and lets b's denote the l'th element and randomness, respectively, in the last list that it got from the Preparer. He checks that blinding with zero (and randomness s) of the "canonical encryption" of b indeed yields the ciphertext d_l. If this is correct, C outputs b [5].

5. Concluding Remarks

Research at the interface between Game Theory and Cryptography is at its infancy and the research area of algorithmic game theory is only about fifteen years old. However, an interesting aspect of our work is the synergy achieved between cryptographic algorithms and game-theoretic problems: By implementing the cryptographic protocol in the game theoretic problem, we gain in the game-theoretic front by eliminating the need for the mediator; we also gain on the cryptography front: for instance, we eliminate the problem of early stopping [5].

6. References

[1] J.F. Nash. Non-Cooperative Games. *Annals of Mathematics,* 54 pages 286–295.

[2] R. Aumann. Subjectivity and Correlation in Randomized Strategies. In *Journal of Mathematical Economics*, 1, pp. 67-95, 1974

[3] J. Garay, R. Gennaro, C. Jutla, and T. Rabin. Secure distributed storage and retrieval. In *Proc. 11th International Workshop on Distributed Algorithms (WDAG '97)*, volume 1320 of *Lecture Notes in Computer Science*, pages 275–289. Springer-Verlag, 1997.

[4] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *27th Annual Symposium on Foundations of Computer Science*, pages 174–187. IEEE, 1986.

[5] Y. Dodis, S. Halevi, and T. Rabin. A Cryptographic Solution to a Game Theoretic Problem. *Lab. of Computer Science,Massachusetts Institute of Technology, 545 Tech Square, Cambridge*, MA 02139, USA.

[6] 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.

[7] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[8] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[9] M. Jakobsson. A Practical Mix. In *Proceedings of EUROCRYPT '98*, pp. 448 461, 1998.

[10] E. Lehrer and S. Sorin. One-shot public mediated talk. Discussion Paper 1108, Northwestern University, 1994.

[11] P. MacKenzie. Efficient ZK Proofs of Knowledge. Unpublished manuscript, 1998.

[12] M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer-Verlag, 1999.

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

[14] A. C. Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, Nov. 1982.

[15] K. Jonathan. Bridging Game Theory and Cryptography: Recent Results and Future Decisions.

[16] D. Chaum and H. Van Antwerpen. Undeniable signatures. In G. Brassard, editor, *Advances in Cryptology — Crypto'89*, pages 212–217, Berlin, 1989. Springer-Verlag. Lecture Notes in Computer Science No. 435.

[17] D. Chaum, C. Cr´epeau, and E. Damg°ard. 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.

[18] D. Chaum and T. Pedersen. Wallet databases with observers. In E. Brickell, editor, *Advances in Cryptology — Crypto'92*, pages 89–105, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science No. 740.

[19] R. Cramer, I. Damgard, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. To appear in *2000 International Workshop on Practice and Theory in Public Key Cryptography*, January 2000, Melbourne, Australia.

[20] C. Cr´epeau and J. Kilian. Weakening security assumptions and oblivious transfer. In *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 2–7. Springer-Verlag, 1990.

[21] C. Dwork, M. Naor, and A. Sahai. Concurrent zero knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 409–418. ACM Press, 1998.

[22] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology – CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer-Verlag, 1985.

[23] I. Barany. Fair distribution protocols or how the players replace fortune. *Mathematics of Operations Research*, 17(2):327–340, May 1992.

[24] J. Gill, "Computational complexity of probabilistic Turing machines." *SIAM Journal on Computing*, 6 (4), pp. 675–695, 1977.