A zero-knowledge Poker protocol that achieves confidentiality of the players' strategy *or* How to achieve an electronic Poker face

Claude Crépeau

Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128 succursale "A", Montréal
Québec, Canada, H3C 3J7

1. Introduction

Many attempts have been previously made to achieve a protocol that would allow people to play mental poker [SRA, GM1, BF, FM, Yu, Cr] (I would rather say electronic poker). But no solution has ever come close to reality with respect to poker strategy. Poker players usually claim that luck has nothing to do with their gains. In fact, poker is a very strategic game. Often, an inexperienced player will loose a lot of money when playing against an experienced player, only because the former cannot hide so easily his emotions. The experienced player can easily know whether his opponent has a good hand or not.

Electronic poker is an ideal medium to hide one's emotions. But, unfortunately, every protocol proposed thus far ruins this perfect poker face since their security is based on the fact that all hands are revealed at the end of the game. This means that the strategy of the players is known to all his opponents. In particular, if one bluffs with a bad hand in the hope that all his opponents will give up, he still has to reveal his hand at the end, in order to participate in the verification part of the protocol. Moreover, when a player opens his hand, he does not want his opponents to learn the moment at which each of his cards was drawn, since this would give them some information about his strategy.

This paper proposes a new poker protocol that allows players to keep secret their strategy. This protocol is an extension of the one given by Crépeau in [Cr]. The security will not be based on the knowledge of the entire deck of card at the end of the game, but rather on some independent information linked to the entries of the deck. This protocol achieves every constraints of a real poker game. It is the first complete solution to the mental poker problem. It achieves all the necessary conditions suggested in [Cr]:

- Uniqueness of cards
- Uniform random distribution of cards
- Absence of trusted third party
- Cheating detection with a very high probability
- Complete confidentiality of cards
- Minimal effect of coalitions
- *Complete confidentiality of strategy*

2. Review of the protocol in [Cr]

Suppose that $P_1,P_2,...,P_N$ want to play poker. Assume a correspondance between the standard deck of cards and the set $\{1,2,...,52\}$. Each P_i will pick a permutation π_i of $\{1,2,...,52\}$ and keep it secret. The shuffled deck will be $\pi_N \cdots \pi_2\pi_1$, i.e.: the functional composition of these permutations. Define $DECK=\{1,2,...,52\}$.

To get a card, player P_i picks a value k in $DECK$ that nobody else has picked before, and will get his card by computing $\pi_N \cdots \pi_2\pi_1(k)$. But since the permutations are kept secret, he will have to use a special trick in order to get this value. To do so, he may use the Hiding-Revealing protocol proposed in [Cr]. This will allow P_i to get the values $\pi_1(k),\pi_2\pi_1(k)$ up to $\pi_N \cdots \pi_2\pi_1(k)$ from his opponents. If everybody was getting their cards this way, all would be fine. But somebody could cheat by computing $\pi_N \cdots \pi_2\pi_1(k')$ for some $k' \in DECK$ not owned by him. This way, he may learn cards which are in the hand of another player or still in the deck. But getting cards that someone else has already is very bad. Obviously, we cannot tolerate this. Unfortunately the protocol of [Cr] solves this problem by asking every player to disclose and prove their π_i at the end of the game, thus revealing every hands, including those of players that would not open their hands at the end of a "physical" Poker game.

How can P_i prove that he is getting a card nobody else has (without revealing this card)? This is the main question addressed (and solved) in this paper.

3. A first idea

To achieve this, we will first change the way by which we check that a player has been reading the entries he claims in his opponents' permutations. The main idea is to add distinct random information to each of the secret values in $\pi_1,\pi_2,..., \pi_N$. This information will be long enough to be hard to guess. When a player reads an entry in the permutation of another player, he will have to read the additional information linked to it. These values will later be publicly revealed by the players, and they all should be different if nobody is cheating.

Let s be a security parameter to be choosen by the players. P_i choses $\tau_{i,j}$, $1 \le j \le i-1$, some arrays of size 52 of distinct bit strings of length s. For $k \in DECK$,

$\tau_{i,j}(k)$ is called the trace of $\pi_i(k)$ for P_j.

To increase the security of π_i we are going to link its traces $\tau_{i,j}$ to it. By linking we mean that the value of $\tau_{i,j}(k)$ will have to be read by player P_j whenever he wants the value $\pi_i(k)$. For this, we use the protocol for the *all-or-nothing disclosure of secrets*, suggested in [BCR], with the 52 secrets

$$<\pi_i(1),\tau_{i,j}(1)>,\ <\pi_i(2),\tau_{i,j}(2)>,\ \cdots,\ <\pi_i(52),\tau_{i,j}(52)>$$

instead of simply using the Hiding-Revealing protocol as before [Cr].

Whenever P_j reads one of the $\pi_i(k)$, he will get the corresponding $\tau_{i,j}(k)$ and he cannot get $\tau_{i,j}(k')$ instead. The interest is that if P_j wants some $\pi_i(k')$ instead of his legitimate $\pi_i(k)$, he will also have to get $\tau_{i,j}(k')$ instead of $\tau_{i,j}(k)$. The point is that if a dishonest player reads an entry owned by P_i, he will get the same trace. We will see later how to use these traces to detect that one of them was cheating, and to figure out which one.

4. All-or-nothing disclosure of secrets (ANDOS)

Let us first see how such a protocol works. Suppose that Alice has a set of t secrets $\{s_1, s_2, \cdots, s_t\}$, and that she wishes to disclose one of them to Bob. Bob does not want Alice to know which secret he takes from the t she has offered him. Alice will choose a secret key for probabilistic encryption, that is two large primes p and q. She will give to Bob the product of them (n) and a quadratic non-residue (y) with Jacobi symbol +1. Let $b_{i,j}$ be the j^{th} bit of the secret s_i. Assume that all the secrets are L bits long. Alice sends to Bob an encrypted version of her secrets. For this, she sends $e_{i,j}$, a random quadratic residue mod n when $b_{i,j}$ is 0 and a random non-residue otherwise.

Now, Bob will build some "questions" about the secrets. To get a secret, Bob will have to ask a question to Alice for each bit of that secret. Typically, a question $Q_{i,j}$ to get bit $b_{i,j}$ looks like $e_{i,j} \times r^2 y^m$ for some randomly selected $r \in \mathbb{Z}_n^*$ and $m \in \{0,1\}$. If Bob asks Alice to decide whether $Q_{i,j}$ is a residue or not, he will be able to compute the value of $b_{i,j}$ since he knows the quadratic relation between $Q_{i,j}$ and $e_{i,j}$. Also, Alice will not have any idea about the bit Bob has been reading since all possible $Q_{i,j}$'s in $\mathbb{Z}_n^*[+1]$ have equal probability, independently of what $e_{i,j}$ is.

When Bob wants a secret, he just asks enough questions to Alice to determine each bit of her secret. But how does Alice know that Bob is not cheating by reading bits in many secrets? He could very well read the first half of some secret together with the second half of another secret.

In order to avoid this, Bob will have to convince Alice that he possesses a set of t fair groups of L questions. A group of questions is fair only if all its questions apply to the same secret. Bob proves to Alice that his groups of questions $Q_{i,\cdot}$ are fair in the way suggested in [BCR]. With this protocol, Bob can convince her that his groups of questions are fair and the probability of doing so when they are not fair is 2^{-s}.

ANDOS PROTOCOL (PREPARATION)

Alice:

STEP 1 chooses p and q, two large primes and computes $n=pq$.

STEP 2 posts n and y, a quadratic non-residue such that $(y/n)=+1$.

STEP 3 chooses $R_{i,j} \in \mathbb{Z}_n^*$ at random for $1 \le i \le t, 1 \le j \le L$.

STEP 4 posts $e_{i,j} = R_{i,j}^2 y^{b_{i,j}}$ **mod** n, a probabilistic encryption of her secrets.

Bob:

STEP 5 chooses ρ a permutation of $\{1,2,...,t\}$.

STEP 6 chooses $r_{i,j} \in \mathbb{Z}_n^*$ and $m_{i,j} \in \{0,1\}$ at random for $1 \le i \le t, 1 \le j \le L$.

STEP 7 posts $Q_{i,j} = e_{\rho(i),j} r_{i,j}^2 y^{m_{i,j}}$.

STEP 8 proves that his groups of questions are fair †.

Whenever Bob wants to get a secret from Alice, he just tells her which group interests him, and she will decide the quadratic character of each question in it. To convince Bob of her fairness, she also sends him a proof of the quadratic residuosity of each question: a square root of Q when Q is a quadratic residue and a square root of Qy when Q is a quadratic non-residue. From this, Bob will be able to compute the value of the secret he wishes, and Alice will be convinced that he is not getting information on more than one secret, but she will not know which secret she gave away.

ANDOS PROTOCOL (GET A SECRET)

Bob:
STEP 1 chooses $i \in \{1,2,...,t\}$ at his will.
STEP 2 sends $\rho^{-1}(i)$ to Alice.

STEP 3 FOR $1 \le j \le L$
Alice:
STEP 3.1 sets $\beta_j = \begin{cases} 0 \text{ if } Q_{\rho^{-1}(i),j} \text{ is a quadratic residue} \\ 1 \text{ otherwise.} \end{cases}$
STEP 3.2 IF $\beta_j = 0$ THEN find r_j such that $r_j^2 \equiv Q_{\rho^{-1}(i),j} \ (mod\ n)$
ELSE find r_j such that $r_j^2 \equiv Q_{\rho^{-1}(i),j}\, y \ (mod\ n)$.
STEP 3.3 sends β_j and r_j to Bob.

Bob:
STEP 3.4 computes $b_{i,j} = \beta_j \oplus m_{\rho^{-1}(i),j}$

5. Some basic difficulties

Since the final solution is still based on the use of permutations, we first consider the problem of proving to the other players that the encrypted string produced by a player is indeed a permutation of {1,2,...,52}. The problem arises from the fact that these permutations must remain secret even after the end of the game. Since they are never opened, they could in fact not be permutations at all.

One might cheat this way, for instance, by pulling out some cards from the deck and replacing them by copies of some other cards. If he does not get caught, he may learn useful information, for instance he may know that no ace of spade exists.

Suppose that P_i wants to use a permutation π_i in the protocol. He would like to convince his opponents that, indeed, π_i is a permutation of {1,2,...,52}. First, he chooses two large primes p_i and q_i and makes public $n_i = p_i q_i$. Next, he chooses and discloses some y_i, a quadratic non-residue modulo n_i with Jacobi symbol +1. Then, he considers each entry of π_i as 6-bit strings and encodes each of them using a random quadratic residue modulo n_i to encode "0"'s and a random quadratic non-residue to encode "1"'s. Let $E_{i,j,k}$ be the encryption of the j^{th} bit of $\pi_i(k)$. Let us call this bit $b_{i,j,k}$. P_i can prove that $E_{i,.,.}$ is the encoding of a permutation of {1,2,...,52} with the following protocol for the values t=52, L=6 and for some given s, a security parameter.

† for details of how the proof is done, see [BCR].

PERMUTATION VALIDITY PROTOCOL

P_i :

STEP 1 chooses $r_{i,j,k} \in \mathbb{Z}_n^*$ at random for $1 \le k \le t, 1 \le j \le L$.

STEP 2 posts $E_{i,j,k} = r_{i,j,k}^2 y^{b_{i,j,\pi_i(k)}}$ **mod** n_i,
a probabilistic encryption of his permutation.

STEP 3 chooses σ_a, for $1 \le a \le s+1$, some permutations of $\{1,2,...,t\}$.

STEP 4 chooses $c_{i,j,k,a} \in \mathbb{Z}_{n_i}^*$ at random for $1 \le k \le t, 1 \le j \le L, 1 \le a \le s+1$

STEP 5 posts $C_{i,j,k,a} = c_{i,j,k,a}^2 y^{b_{i,j,\sigma_a(k)}}$ **mod** n_i.

STEP 6 Altogether the players pick s random bits $i_2, i_3, \ldots, i_{s+1}$.

STEP 7 FOR each $i_a = 1$

STEP 7.1 P_i reveals $r_{i,j,\pi_i^{-1}(k)} c_{i,j,\sigma_a^{-1}(k),a}$ for $1 \le k \le t, 1 \le j \le L$.

STEP 8 FOR each $i_a = 0$

STEP 8.1 P_i reveals $c_{i,j,\sigma_1^{-1}(k),1} c_{i,j,\sigma_a^{-1}(k),a}$ for $1 \le k \le t, 1 \le j \le L$.

STEP 9 P_i reveals $c_{i,j,k,1}$ for $1 \le k \le t, 1 \le j \le L$
to prove that $C_{i,.,.,1}$ encrypts a permutation of $\{1,2,...,t\}$.

For further details on the construction of this protocol, see [BC]. P_i will be able to prove to his opponents that $E_{i,.,.}$ is the encoding of a permutation of $DECK$, when in fact it is not, with probability 2^{-s}. In fact, STEPs 1 to 8.1 proves that $E_{i,.,.}$ and $C_{i,.,.,1}$ encrypts two permutations of the same set, and STEP 9 reveals that set.

But in order for this protocol to work, n_i must be of the adequate form (with only two prime factors). In fact, the protocol works whenever $n_i = p_i^a q_i^b$ with both p_i and q_i dinstinct primes and a and b not both even. In order to prove that n_i is of the correct form, P_i may use the protocol given in [GHY]. By repeating this protocol, P_i can convince his opponents that n_i is of the good form. Also, to prove that y_i is a quadratic non-residue modulo n_i he can use the protocol given in [GMR].

Notice that all the protocols suggested so far are zero-knowledge. Initially, each player P_i uses PREPARATION(i) as suggested below:

PREPARATION(i)

P_i:

STEP 1 chooses π_i, a permutation of $DECK$.

STEP 2 chooses $\tau_{i,j}$, for $1 \le j \le i-1$,
some arrays of 52 random distinct strings of $\{0,1\}^s$.

STEP 3 chooses p_i, q_i, y_i
as in both ANDOS PREPARATION
and PERMUTATION VALIDITY protocols.

STEP 4 posts $n_i = p_i q_i$ and y_i.

STEP 5 proves that n_i and y_i are in the correct form.

STEP 6 reveals probabilistic encryptions of π_i and each $\tau_{i,j}$
for $1 \le j \le i-1$.

STEP 7† uses ANDOS PREPARATION PROTOCOL with P_j
for the secrets $<\pi_i(1), \tau_{i,j}(1)>$, ..., $<\pi_i(52), \tau_{i,j}(52)>$.

STEP 8 P_i proves that π_i is indeed a permutation of $DECK$,
using PERMUTATION VALIDITY PROTOCOL from
STEP 3

6. Using the traces

We must now see how these traces are useful in the detection of cheaters. The fundamental properties of the traces is: $\forall i \, \forall j \, \forall k, k' \quad \pi_i(k) \ne \pi_i(k') \Longrightarrow \tau_{i,j}(k) \ne \tau_{i,j}(k')$

The $\tau_{i,j}$'s will enable the opponents P_j of P_i, for $j > i$, to convince him that they have not been reading some of his own entries in π_i. But first let us see how an honest player gets a card.

Initially, each number k in $DECK$ is marked "free". To get a new card, player P_i picks a "free" value k and mark it "used". We say that k is the identifier of the card. Then, P_i will ask publicly his opponents for the values of $\pi_1(k)$, $\pi_2\pi_1(k)$ up to $\pi_{i-1} \cdots \pi_1(k)$. They will prove their claims by decrypting the corresponding entries of their coded permutations (by decrypt we mean that they reveal the random seed they have used to encrypt that information). Then P_i gets $\pi_i \pi_{i-1} \cdots \pi_1(k)$ by looking at his own permutation. Finally he gets the values $\pi_{i+1} \cdots \pi_1(k)$ up to $\pi_N \cdots \pi_1(k)$ by using the secret questions he has proven correct to P_{i+1}, P_{i+2}, ..., P_N. When he does this, he also gets the corresponding strings $\tau_{i+1,i} \pi_i \cdots \pi_1(k)$ up to $\tau_{N,i} \pi_{N-1} \cdots \pi_1(k)$. These strings will allow him to prove later that he was honest when reading in π_{i+1}, π_{i+2}, ..., π_N.

† start at STEP 6 of ANDOS PREPARATION with Alice = P_i and Bob = P_j

GET A CARD(i)

STEP 1 P_i picks k a free value in $DECK$; marks it used.
STEP 2 sets $c = k$
STEP 3 FOR p = 1 TO $i-1$
STEP 3.1 P_i gets $\pi_p(c)$ from P_p (publicly)
STEP 3.2 sets $c = \pi_p(c)$
STEP 4 P_i sets $c = \pi_i(c)$
STEP 5 FOR p = i+1 TO N
STEP 5.1 P_i gets $<\pi_p(c), \tau_{p,i}(c)>$ using the ANDOS GET A SECRET PROTCOL
STEP 5.2 sets $c = \pi_p(c)$
STEP 6 CARD = c

Later in the game, the players will have to prove each other that they have not been cheating by reading some of the cards of their opponents. In fact, we can tolerate that a player reads a card which does not belong to him as long as this card does not belong to someone else, because this does not change the distribution of probability of the hands of the players. Getting any "free" card is equivalent. The only trouble in this case is that the lucky cheater (lucky because he won't get caught) will not be able to use this card since he cannot prove he read it honestly.

In general, if X is a set and f a function over the elements of that set, we use

$$f(X) = \{y \mid y = f(x) \text{ and } x \in X\}.$$

Let $H_{i,j}$ denote the set of values of $DECK$ owned by P_j in the permutation π_i. $H_{0,j}$ is the set of identifiers P_j has picked in order to get his cards.

To be fair with P_i, a player P_j must have gotten $\pi_i(H_{i-1,j})$ and $\tau_{i,j}(H_{i-1,j})$. Whenever P_i wants to make sure that nobody has read one of his entries he may proceed in the following way. Note that $\bigcup_{j \geq i} H_{i-1,j}$ is publicly known. First, P_i reveals $\tau_{i,k}(\bigcup_{j \geq i} H_{i-1,j})$ for $1 \leq k \leq i-1$. He then asks to each of his opponents P_j such that $j < i$ (those who have been reading in his π_i with ANDOS) to reveal an encrypted version of $\tau_{i,j}(H_{i-1,j})$ and they do so. Then P_i reveals and decrypts $\tau_{i,k}$ for $1 \leq k \leq i-1$. Now, P_j reveals an encryption for the entries of $\tau_{i,j}$ of which nobody has yet claimed property $(\tau_{i,j}(DECK) - \tau_{i,j}(H_{i-1,j}) - \tau_{i,k}(\bigcup_{j \geq i} H_{i-1,j}))$. He then proves that, altogether, the encrypted strings he posts, constitute a permutation of $\tau_{i,j}(DECK) - \tau_{i,k}(\bigcup_{j \geq i} H_{i-1,j})$, i.e. none of these strings is own by any of P_1 to P_{j-1}. For this they use the PERMUTATION VALIDITY PROTOCOL (except STEPs 1 and 2).

This process enables P_j to prove that, indeed, he has been reading some strings from $\tau_{i,j}$ and that they are not among $\tau_{i,j}(\bigcup_{k \geq i} H_{i-1,k})$.

DETECTION OF CHEATERS

STEP 1 FOR $p = 2$ TO N
STEP 1.1 FOR $i = 1$ TO $p-1$
STEP 1.1.1 P_i sets secretly $T_i = \tau_{p,i}(H_{p-1,i})$
STEP 1.1.2 P_i displays publicly E_i, a probabilistic encryption of $T_{p,i}$

STEP 1.1.3 P_p posts publicly $\tau_{p,i}$
STEP 1.1.4 P_p displays publicly $T_p = \tau_{p,i}(\bigcup_{j \geq i} H_{p-1,j})$

STEP 1.1.5 P_i sets secretly $T'_i = \tau_{p,i}(DECK) - T_i - T_p$.
STEP 1.1.6 P_i displays publicly E'_i a probabilistic encryption of T'_i.
STEP 1.1.7 P_i proves that $E_i \cup E'_i$ is
an encrypted permutation of $\tau_{p,i}(DECK) - T_p$
using the PERMUTATION VALIDITY PROTOCOL.

STEP 2 IF cheating occured THEN KICK OUT cheaters.

Notice that this complete process may be performed at any moment, whenever a player wants to check the honesty of his opponents. The normal situation can be restored by generating new arrays $\tau_{i,j}$ to replace the previous ones. P_i creates a new $\tau_{i,j}$ and posts an encrypted version of it in order to link it back with π_i.

At this point, the players will return into the process of generation of fair groups of questions about $<\pi_i, \tau_{i,j}>$ as in the preparation protocol. These new questions should match exactly with the previous ones in the π_i part in order to know the questions that were used before the DETECTION OF CHEATERS protocol.

Once every player P_j, $j < i$, has convinced P_i that they have a set of fair groups of questions, he will allow each of them to read $\#H_{i-1,j}$ entries of $\tau_{i,j}$ in order for them to update the values of strings they should know ($\tau_{i,j}(H_{i-1,j})$). These strings should correspond to those he knew before executing the DECTECTION OF CHEATERS protocol. Notice that if we had not forced the π_i part to match, then P_i would not know if P_j was reading the same entries he had before.

TRACES RESTORATION(i)

P_i:

STEP 1 chooses new $\tau_{i,j}$, for $1 \le j \le i-1$,
some arrays of 52 random distinct strings of $\{0,1\}^s$.

STEP 2 reveals a probabilistic encryption of each $\tau_{i,j}$ for $1 \le j \le i-1$.

STEP 3‡ uses ANDOS PREPARATION PROTOCOL with P_j for $1 \le j \le i-1$
for the secrets $<\pi_i(1),\tau_{i,j}(1)>$, ..., $<(\pi_i(52),\tau_{i,j}(52)>$.

STEP 4 allowes P_j to read exactly $\tau_{i,j}(H_{i-1,j})$
by using the ANDOS GET A SECRET PROTOCOL

Generally, these two protocols should be requested by P_i before he makes any strategic moves, such as betting or declaring the number of cards he wishes to discard. Otherwise, some cheater may learn some strategic information before being detected and communicate it to the remaining players.

7. Opening and Closing of hands.

We have not yet discussed the way by which the players will open a card or declare it closed for the rest of the game (discarded). One might think that claiming "I discard k" for some identifier k that I own, should be sufficient to discard a card. In the same way, maybe, it would be fine to open a card to reveal $\pi_i \pi_{i-1} \cdots \pi_1(k)$, $\pi_{i+1}\pi_i \cdots \pi_1(k)$, ..., $\pi_N \pi_{N-1} \cdots \pi_1(k)$.

But this way, some strategic information will be acquired by the players about their opponents. Suppose that my hand includes the cards of figure 1 (below). Then I may discard the first 2 cards and draw 2 new ones. Suppose I then get into the situation of figure 2.

If I open up my hand according to the above described protocol, my opponents would know which of my cards are the new ones. This way, they may learn information about my strategy.

‡ start at STEP 6 of ANDOS PREPARATION, with Alice=P_i, Bob=P_j. Note that the $r_{i,j}$ used as random seed should not change for the π_i part of the secrets. This will make a new set of groups of questions that matches exactly the previous one on the π_i part.

To solve this problem, the players will carry an encrypted permuted version of their hand for the entire game. In fact, each player P_i will manage publicly an encrypted version of $H_{i,i}$. Note that this information is sufficient to determine his hand. Define $D_i \subseteq H_{i,i}$ as the subset of values in $H_{i,i}$ which are leading to a discarded card. Clearly, $H_{i,i} - D_i$ is the subset of $H_{i,i}$ with elements leading to a card of P_i's hand.

Initially, $H_{i,i}$ and D_i are empty. Whenever P_i gets a card with identifier k, he places the encryption of $\pi_i \pi_{i-1 \ldots} \pi_1(k)$ into an encrypted version of $H_{i,i}$. Before opening or discarding a card, he will confuse his opponents about the origin of the cards in $H_{i,i}$ by generating a new coded permutation of the elements in $H_{i,i} - D_i$ and prove it so with the PERMUTATION VALIDITY PROTOCOL. He will then use this new $H_{i,i}$ to make his operation. The point is that his opponents are convinced that $H_{i,i}$ still includes the same elements, but they no longer know in which order. Moreover they know that D_i has not changed.

If P_i wants to discard a card from his hand, he transfers the corresponding element of $H_{i,i} - D_i$ into D_i using its encrypted version. On the other hand, if he wants to open it, he just decrypts the corresponding entry of $H_{i,i} - D_i$ and uses it to follow the corresponding values in π_i, π_{i+1}, ..., π_N in order to get to his card. (remember that the values in $H_{i,i}$ are of the form $\pi_i \pi_{i-1 \ldots} \pi_1(k)$). Define $\hat{H}_i$ as the probabilistic encrypted version of $H_{i,i}$. Similarly define $\hat{D}_i$ from D_i.

DISCARD(i,k)

STEP 1 P_i places the entry of $\hat{H}_i$ corresponding to $\pi_i \pi_{i-1} \cdots \pi_1(k)$ into $\hat{D}_i$.

OPEN A CARD(i,k)

STEP 1 set $c = \pi_i \pi_{i-1} \cdots \pi_1(k)$
STEP 2 P_i reveals c
STEP 3 P_i decrypts the entry of $\hat{H}_i - \hat{D}_i$ corresponding to c.
STEP 4 FOR p = i+1 TO N
STEP 4.1 P_i reveals $\pi_p(c)$ and $\tau_{p,i}(c)$
STEP 4.2 P_p decrypts $\pi_p(c)$ and $\tau_{p,i}(c)$.
STEP 4.3 set $c = \pi_p(c)$

8. General protocol

Finally, here is how all these ideas fit together in order to accomplish a fair, purely secure, game of electronic poker:

POKER PROTOCOL

STEP 1 each player P_i uses PREPARATION(i)

STEP 2 REPEAT
STEP 2.1 each P_i gets his cards using GET A CARD(i)
STEP 2.2 each P_i adds the encryption of $\pi_i \pi_{i-1} \cdots \pi_1(k)$ to $\hat{H}_i$ each time he picks a card starting from the identifier k.

Altogether the players:
STEP 2.3 use DECTECTION OF CHEATERS

STEP 2.4 each P_i generates a new permuted version of $\hat{H}_i - \hat{D}_i$ and
proves it by using
PERMUTATION VALIDITY PROTOCOL
(except STEPs 1,2 and 8).

According to the rules and to their strategic decisions, the players:
STEP 2.5 bet, discard and open some cards
using DISCARD and OPEN A CARD.

STEP 2.6 IF NOT the end of the game
STEP 2.6.1 each P_i uses TRACES RESTAURATION(i)

STEP 3 UNTIL the end of the game

9. In conclusion

We have achieved the first complete solution to the mental poker problem. Our solution cumulates all the conveniences of a real poker game *and* the elimination of the unfortunate human factor (from a strategic point of view). In order to solve even more problems of card playing or similar games (such as Scrabble), with special operations such as returning cards into the deck, the full power of Boolean circuit simulation suggested in [BC] can be used. But unfortunately, the resulting protocol is too messy to be explained here.

10. ACKNOLEDGEMENTS

I wish to tank Gilles Brassard for the numerous discussions we had about the protocols and for the large amount of work he did with me [BCR, BC]. I would like also to thank Roxane for her support.

11. REFERENCES

[BF] Banary, I. and Füredi, Z. ''Mental Poker with Three or More Players'' in *Information and Control*, 59 (1983), pp. 84-93.

[BC] Brassard, G. and Crépeau C., ''Zero-Knowledge Simulation of Boolean Circuits'', presented at CRYPTO 86.

[BCR] Brassard G., Crépeau C. and Robert J.-M., ''All-or-Nothing Disclosure of Secrets'', presented at CRYPTO 86.

[Cr] C. Crépeau, ''A Secure Poker Protocol That Minimizes the Effect of Player Coalitions'', *Advances in Cryptology: Proceedings of CRYPTO 85*, H. C. Williams ed., Lecture Notes in Computer Science 218, Springer-Verlag, Berlin, 1986, pp73-86.

[FM] Fortune, S. and Merrit, M., ''Poker Protocols'' in *Advances in Cryptology: Proc. of CRYPTO 84*, G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp.454-464.

[GHY] Galil, Z., S. Haber and M. Yung, ''A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge Public-Key Cryptosystems'' *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science*, 1985, pp. 360-371.

[GM1] Goldwasser, S. and Micali S., ''Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information'' in *Proceedings of the 14th Annual ACM symp. on Theory of computing*, ACM-SIGACT, May 1982, pp. 365-377.

[GM2] Goldwasser, S. and Micali S., ''Probabilistic Encryption'' in *J. Comput. System Sci.*, 28 (1984), pp. 270-299.

[GMR] Goldwasser, S., S. Micali and C. Rackoff, ''The Knowledge Complexity of Interactive Proof-Systems'' *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, 1985, pp. 291-304.

[SRA] Shamir, A., Rivest R. and Adleman L., ''Mental Poker'' MIT Technical Report, 1978.

[Yu] Yung, M., ''Cryptoprotocols: Subscription to a Public Key, The Secret Blocking and the Multi-Player Mental Poker Game'' in *Advances in Cryptology: Proc. of CRYPTO 84*, G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp.439-453.