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

Claude Crépeau

Laboratory for Computer Science
M.I.T.
545 Technology Square
Cambridge Massachusetts 02141 USA

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). Unfortunatly 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 way of hiding one's emotions. But, in fact, 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*

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 239-247, 1987.

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 $DECK = \{1,2,...,52\}$. Each P_i will pick a permutation π_i of $DECK$ and keep it secret. The shuffled deck will be $\pi_N \cdots \pi_2 \pi_1$, i.e.: the functional composition of these permutations.

To get a card, player P_i picks a value k in $DECK$ that nobody else has picked before, and gets his card by computing $\pi_N \cdots \pi_2 \pi_1(k)$. 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 $k' \in DECK$ he does not own. This way, he may learn cards which are in the hand of another player or still in the deck. Obviously, we cannot tolerate that he gets cards that someone else has already picked. Unfortunately the protocol of [Cr] solves this problem by asking every player to disclose 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 "real" 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 some random information to each of the secret values in $\pi_1, \pi_2, ..., \pi_N$. This information will be randomly chosen bit strings which are 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 bit string linked to it. These strings will later be publicly revealed by the players who wish to open their hands, and they all should match the initial strings if nobody is cheating.

Let s be a security parameter to be chosen by the players. P_i chooses $\tau_i : DECK \rightarrow \{0,1\}^s$. For $k \in DECK$, the string $\tau_i(k)$ is called the trace of $\pi_i(k)$.

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

$$<\pi_i(1),\tau_i(1)>, \; <\pi_i(2),\tau_i(2)>, \; \cdots, \; <\pi_i(52),\tau_i(52)>$$

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 $\tau_i(k)$ and he cannot get $\tau_i(k')$ 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 $\tau_i(k')$ instead of $\tau_i(k)$. Later in the game he will not be able to convince his opponents that he has read $\pi_i(k)$ since he do not know $\tau_i(k)$ and can guess it only with a very small probability.

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 $\hat{b}_{i,j}$, a random quadratic residue mod n when $b_{i,j}$ is 0 and a random non-residue otherwise.

ANDOS PROTOCOL (Encryption of Secrets)

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 $1 \leq i \leq t, 1 \leq j \leq L$.**
STEP 4 **posts $\hat{b}_{i,j} = R_{i,j}^2 y^{b_{i,j}}$ mod n, a probabilistic encryption of her secrets.**

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 $\hat{b}_{i,j} \times r^2 y^m$ 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 $\hat{b}_{i,j}$. 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 $\hat{b}_{i,j}$ 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 $Q_{i,}$ 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 achieving such a proof when they are not fair is 2^{-s}.

ANDOS PROTOCOL (Preparation of Questions)

Bob:

STEP 1 **chooses ρ a permutation of $\{1,2,...,t\}$.**
STEP 2 **chooses $r_{i,j} \in \mathbb{Z}_n^*$ and $m_{i,j} \in \{0,1\}$ at random for $1 \leq i \leq t, 1 \leq j \leq L$.**
STEP 3 **posts $Q_{i,j} = \hat{b}_{\rho(i),j} r_{i,j}^2 y^{m_{i,j}}$.**
STEP 4 **proves that his groups of questions are fair (see [BCR] for details).**

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 $1 \leq j \leq L$
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    finds $r_j$ such that $r_j^2 \equiv Q_{\rho^{-1}(i),j} y^{\beta_j} \ (mod\ n)$.
STEP 3.3    sends $\beta_j$ 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 $\{1,2,...,52\}$. For this he can use a general purpose protocol proving that two encrypted permutations contain the same set of elements. So, we therefore consider first the implementation of this general protocol.

Let $X = \{x_1, x_2, \cdots, x_t\}$ be a set known to Bob. Let σ and σ' be two permutations of the elements of X. Consider x_i as a bit string of length L, where $L = \max\{|x_i| : 1 \leq i \leq t\}$. Define $b_{i,j}$ to be the j^{th} bit of x_i. Let $(n = pq, y)$ be Bob's probabilistic encryption public keys. Finally, let $\hat{b}_{i,j}$ be a probabilistic encryption of $b_{\sigma(i),j}$ and $\hat{b}'_{i,j}$ be a probabilistic encryption of $b_{\sigma'(i),j}$.

PERMUTATION EQUALITY PROTOCOL (Preparation)

```
Bob:
STEP 1  chooses $r_{i,j}, r'_{i,j} \in \mathbb{Z}_n^*$ at random for $1 \leq i \leq t, 1 \leq j \leq L$.
STEP 2  posts $\hat{b}_{i,j} = r_{i,j}^2 y^{b_{\sigma(i),j}} \bmod n$ and $\hat{b}'_{i,j} = r'^2_{i,j} y^{b_{\sigma'(i),j}} \bmod n$,
        some probabilistic encryptions of his permutations.
```

Bob can then prove in zero-knowledge to Alice that for all i their exists i' such that for all j $\hat{b}_{i,j}$ and $\hat{b}'_{i',j}$ encrypt the same bit, using the following protocol. (thus proving that the $\hat{b}_{i,j}$'s and the $\hat{b}'_{i,j}$'s encrypt permutations of the same set). Let s be a security parameter agreed between Bob and Alice.

PERMUTATION EQUALITY PROTOCOL

```
STEP 1  FOR $1 \leq k \leq s$
    STEP 1.1  Bob chooses $\rho$, a random permutations of $\{1,2,...,t\}$.
    STEP 1.2  Bob chooses $c_{i,j} \in \mathbb{Z}_n^*$ at random for $1 \leq i \leq t, 1 \leq j \leq L$
    STEP 1.3  Bob posts $\bar{b}_{i,j} = c_{i,j}^2 y^{b_{\rho(i),j}} \bmod n$.
    STEP 1.4  Alice chooses a bit $c$ at random and tells it to Bob.
    STEP 1.5  IF $c=0$ Bob reveals $r_{\sigma^{-1}(i),j,} c_{\rho^{-1}(i),j}$ for $1 \leq i \leq t, 1 \leq j \leq L$.
    STEP 1.6  IF $c=1$ Bob reveals $r'_{\sigma'^{-1}(i),j} c_{\rho^{-1}(i),j}$ for $1 \leq i \leq t, 1 \leq j \leq L$.
```

For further details on the construction of this protocol, see [BC]. *Bob* will be able to prove to Alice that $\hat{b}_{i,.}$ and $\hat{b}'_{i,.}$ are encoded permutations of the same set, when in fact it is not, with probability 2^{-s}.

In our case, P_i simply uses $\sigma = \pi_i$ and $\sigma' = I$, where I is the identity permutation. Once the protocol is completed, P_i decrypts the $\hat{b}'_{i,j}$'s and prove that they constitute an encryption of I(by decrypting we mean that he reveals the random seed used to encrypt that information) . The preparation part of the protocol may be performed only once, while the second part of the protocol should be performed with each opponent separetly. Of course, each player P_i uses his personal values n_i and y_i in place of n and y in the previous protocol.

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 each of 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 (under the assumption that deciding quadratic residuosity is hard). This makes the following preparation protocol zero-knowledge. Initially, each player P_i uses PREPARATION(i) as suggested below:

```
PREPARATION(i)

$P_i$:
STEP 1    chooses $\pi_i$, a permutation of $DECK$.
STEP 2    chooses $\tau_i : DECK \to \{0,1\}^s$ at random.
STEP 3    chooses $p_i$, $q_i$ and posts $n_i = p_i q_i$ and $y_i$.
STEP 4    proves that $n_i$ and $y_i$ are in the correct form.
STEP 5    reveals probabilistic encryptions of $\pi_i$ and $\tau_i$
STEP 6    uses ANDOS PROTOCOL (Preparation of Questions) with
          each $P_j$ for the secrets $<\pi_i(1),\tau_i(1)>$, ..., $<\pi_i(52),\tau_i(52)>$.
STEP 7    $P_i$ proves that $\pi_i$ is indeed a permutation of $DECK$,
          using PERMUTATION EQUALITY PROTOCOL.
```

6. Getting cards

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 asks 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. 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 $\tau_{i+1}\pi_i \cdots \pi_1(k)$ up to $\tau_N\pi_{N-1} \cdots \pi_1(k)$. These strings will allow him to prove later that he was honest when reading in π_{i+1} , π_{i+2}, ..., π_N.

```
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[†]   $P_i$ adds $\hat{\pi}_i(c)$ to $\hat{H}_i$
STEP 5      $P_i$ sets $c = \pi_i(c)$
STEP 6      FOR $p = 1$ TO $i-1$
STEP 6.1        $P_p$ shows that he has never used his group of
                questions that could read $\pi_i(c)$.
STEP 7      FOR $p = i+1$ TO $N$
STEP 7.1        $P_i$ gets $<\pi_p(c),\tau_p(c)>$
                using the ANDOS PROTOCOL (Get a Secret)
STEP 7.2        sets $c = \pi_p(c)$
STEP 8      CARD $= c$
```

[†] The meaning of this step will become clear in the next section.

This protocol tolerate that a player reads a card which does not belong to him but only if 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.

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 $\pi_i\pi_{i-1}\cdots\pi_1(k)$, $\pi_{i+1}\pi_i\cdots\pi_1(k)$, ..., $\pi_N\pi_{N-1}\cdots\pi_1(k)$ (since $\pi_1(k)$, $\pi_2\pi_1(k)$, ..., $\pi_{i-1}\pi_{i-2}\cdots\pi_1(k)$ are already known publicly).

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.

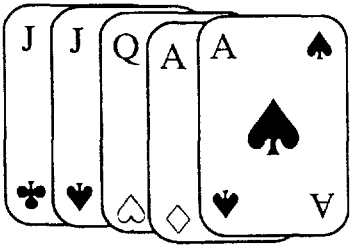

figure 1

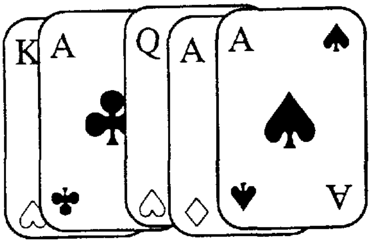

figure 2

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.

Let K_i denote the set of values of *DECK* owned by P_i in his own permutation π_i. To solve the above mentionned problem, the players will carry an encrypted permuted version of their K_i for the entire game. Note that this information is sufficient to determine his hand. Define $D_i \subseteq K_i$ as the subset of values in K_i which are leading to a discarded card. Clearly, $H_i = K_i - D_i$ is the subset of K_i with elements leading to a card of P_i's hand.

Initially, H_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 $\hat{H}_i$, an encrypted version of H_i. Before opening or discarding a card, he will confuse his opponents about the origin of the cards in H_i by generating a new encrypted permutation of the elements in H_i and prove it so with the PERMUTATION EQUALITY PROTOCOL. He will then use this new $\hat{H}_i$ to make his operation. The point is that his opponents are convinced that H_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 $\hat{H}_i$ into $\hat{D}_i$.

```
DISCARD(i,k)

STEP 1  $P_i$ generates a new permuted version of $\hat{H}_i$ and
        uses PERMUTATION EQUALITY PROTOCOL
        to prove that $H_i$ has not changed.
STEP 2  $P_i$ places the entry of $\hat{H}_i$ corresponding to $\pi_i\pi_{i-1} \cdots \pi_1(k)$ into $\hat{D}_i$.
```

On the other hand, if he wants to open it, he just decrypts the corresponding entry of $\hat{H}_i$ 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 K_i are of the form $\pi_i\pi_{i-1\ldots}\pi_1(k)$).

```
OPEN A CARD(i,k)

STEP 1  $P_i$ generates a new permuted version of $\hat{H}_i$ and
        uses PERMUTATION EQUALITY PROTOCOL
        to prove that $H_i$ has not changed.
STEP 2  set $c=\pi_i\pi_{i-1} \cdots \pi_1(k)$
STEP 3  $P_i$ reveals $c$
STEP 4  $P_i$ decrypts the entry of $\hat{H}_i$ corresponding to $c$.
STEP 5  FOR $p$ = i+1 TO $N$
STEP 5.1    $P_i$ reveals $\pi_p(c)$ and $\tau_p(c)$
STEP 5.2    $P_p$ decrypts $\hat{\pi}_p(c)$ and $\hat{\tau}_p(c)$.
STEP 5.3    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 UNTIL the end of the game
STEP 2.1    each $P_i$ gets his cards using GET A CARD($i$)
According to the rules and to their strategic decisions, the players:
STEP 2.2    bet, discard and open some cards
            using DISCARD and OPEN A CARD.
```

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. ACKNOWLEDGEMENTS

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 Furedi, 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.