CRYPTOPROTOCOLS: SUBSCRIPTION TO A PUBLIC KEY, THE SECRET BLOCKING AND THE MULTI-PLAYER MENTAL POKER GAME (extended abstract)

Mordechai Yung

CUCS-137-84

CUCS-137-84

CRYPTOPROTOCOLS: SUBSCRIPTION TO A PUBLIC KEY, THE SECRET BLOCKING AND THE MULTI-PLAYER MENTAL POKER GAME

(extended abstract)

Mordechai Yung

Department of Computer Science

Columbia University

New York, N.Y. 10027

ABSTRACT

Investigating the capabilities of public key and related cryptographic techniques has recently become an important area of cryptographic research. In this paper we present some new algorithms and cryptographic protocols (*Cryptoprotocols*) which enlarge the range of applications of public key systems and enable us to perform certain transactions in communication networks. The basic cryptographic tools used are Rabin's *Oblivious Transfer Protocol* and an algorithm we developed for *Number Embedding* which is provably hard to invert.

We introduce the protocol *Subscription to a Public Key*, which gives a way to transfer keys over insecure communication channels and has useful applications to cryptosystems. We develop the *Secret Blocking Protocol*, specified as follows : 'A transfers a secret to B, B can block the message. If B does not block it, there is a probability P that he might get it. ($1/2 \leq P < 1$, where we can control the size of P). A does not know if the message was blocked (but he can find out later)'.

The classic cryptotransaction is the *Mental Poker Game*. A cryptographically secure solution to the *Multi Player* Mental Poker Game is given. The approach used in constructing the solution provides a general methodology of provable and modular *Protocol Composition*.

1. INTRODUCTION

Complexity-based cryptography has two major areas of application: *Public Key Cryptosystems* [7] [15], to provide secure and authenticated communication, and Cryptographic Transactions, *Cryptotransactions* for short, to enable simulation of certain activities in communication [4] [6] [17]. These activities, while easily done face to face seem impossible to perform through the use of a communication network.

In this paper we present some new Cryptoprotocols to be used both for increasing security and flexibility of Public-Key Cryptosystems and as tools for implementing Cryptotransactions. The security of these protocols is based on the intractability of the factorization problem. The basic cryptographic tools used are Rabin's *Oblivious Transfer Protocol* and an algorithm for *number embedding* which is provably hard to invert. The results reported here were motivated by Blum's paper [4] and are based on [19].

We introduce the protocol *Subscription to a Public Key*, used for transferring keys over insecure communication channels and which has useful applications for cryptosystems. We then develop the *Secret Blocking Protocol*, specified as follows : "A transfers a secret to B. B can block the message. If B does not block the message he gets it with probability = P, where $1/2 \leq P < 1$, and we can control the size of P. A does not know if the message was blocked, but he can find out later".

The classic cryptotransaction is the *Mental Poker Game*. The problem, proposed by Robert Floyd, is: 'Is it possible to play a fair poker game over the telephone ?' Shamir, Rivest and Adleman [17] proved that from an information theoretic point of view it is impossible to play the game. They showed, however that from a complexity theoretic point of view, the game can be played, using the one way commutative modular exponentiation function. Although their protocol is elegant and the number of players is unlimited, Lipton [10] showed that one can easily mark some subsets of cards using it. We present a cryptographically secure solution to the *Multi Player Mental Poker Game*. Different solutions to the Two Player Mental Poker Game have recently been obtained independently by Blum [5], and by Goldwasser and Micali [8]. Their solutions include a protocol for Two Player Card Dealing.

The approach used in constructing the solution gives a general methodology of provable and modular *Protocol Composition*.

2. Number Theoretic and Cryptographic Background

2.1. NUMBER THEORETIC ALGORITHMS AND PUBLIC KEY SYSTEMS

The *main assumption* is: *FACTORIZATION* of a number n=p q, where p and q are large (say 100-digit) prime numbers is HARD to solve. On the other hand, some number theoretic algorithms that we use are EASY (random polynomial time). These include the primality test, [18] [9], prime generation and root extraction of x^2(modn) given the factors of n [12]. (For a survey of number theory and number theoretic algorithms see [11] [9] [1].) In the protocols presented in this paper, we need an underlying public key system in order to transmit encoded and signed messages and to hide information using one way functions. Either the RSA system [15], the Rabin system [12], or the Blum-Goldwasser system [3] can be used. (If we use RSA we add the assumption that RSA breaking is HARD)

2.2. RABIN'S OBLIVIOUS TRANSFER PROTOCOL

Rabin found a way to send a secret obliviously, that is, the sender A does not know if the secret is successfully transmitted to B, the probability of success is 1/2.

Protocol 1 -THE OBLIVIOUS TRANSFER:
step 1 : A creates a number n =p q. (The prime factorization of n is the secret.)
step 2 : A-->B : " n". (--> means 'sends to'.)
step 3 : B selects a random x and computes z = x^2(mod n). B-->A : " z".
step 4 : A, knowing the factorization of n, computes the 4 square roots of z = { x,-x,y,-y}. He selects at random one of them, calls it s and A-->B:" s"
step 5 : If B receives y or -y he gets the secret, if he receives x or -x he does not.
end {protocol 1}

> **Theorem 1:** Given $x,y \in Z_n^*$ (that is x,y<n and do not divide n), $x \neq y$ (mod n), $-x \neq y$ (mod n) and $x^2 = y^2$ (mod n), there is an EASY algorithm for factoring n.

Based on the previous theorem we can prove the properties of the above protocol:

> **Theorem 2:** Using the oblivious transfer protocol, B can factor n (get the secret) with a probability (virtually) equal to 1/2. A can not know if he transferred the secret successfully.

2.3. ONE WAY NUMBER EMBEDDING

Number embedding is an algorithm which gets a number M as input and distributes it into some pieces of information EM(M) which hide M. Giving EM(M) does not compromise M because in order to recover the number from the available hiding information one has to solve a HARD problem. EM(M) can be recovered to one and only one number: M. In [4] Blum gave such an algorithm. Here we use polynomial interpolation to design a number embedding algorithm that is provably

HARD to recover.

Algorithm 2 - EMBEDDING USING INTERPOLATION :
step 1: Choose K (say K=10) random 100-digit prime numbers p_i, i=1,..,10.
step 2: Choose 10 random 99-digit prime numbers q_i, i=1,..,10.
step 3: Construct a polynomial of degree 10: P (mod R), where R is a large prime ($R > p_i, q_i$ i=1,..,10), by using the 11 interpolation points : (p_i, q_i) i=1,..,10, and (0,M).
step 4: Compute $n_i = p_i\ q_i$ (n_i hides interpolation point i). The embedding of M is the sequence consisting of: R (the modulus), the numbers n_i, i=1,..,10, and a point (u,v) such that v=P(u), where u is a random number different from 0 and the p_i's.
end {algorithm 2}

The Result of the Algorithm:
Given EM(M), one has to factor the 10 n_i's to recover the unknown M. Factorization of any 9 of them does not help (see [16]). Given $M_1 \neq M_2$, we can embed both using the same n_i's; only the additional random point (u,v) is different. Therefore we can prepare all the n_i's before the communication. EM is a one way one to many random operator. Using the fact that generation of numbers of the form n=p q is easy, and the random polynomial algorithm [2] [13] for finding roots of polynomials over GF(R) we can show that recovering of M is polynomially equivalent to factorization. The reduction to factorization is given in the following theorem:

> **Theorem 3:** If we can easily recover M from EM(M) (even in ϵ of the times) we can easily factor numbers of the form n=p q.

Now consider the oblivious transfer protocol. If we want A to be able to check whether or not he gave B the secret then we use ***Oblivious Transfer With Receipt***: When B sends $z = x^2$ (mod n) he also sends EM(x) which is the receipt which hides x unambiguously. The receipt also makes it possible to check that z was created by squaring an $x \in Z_n^*$ and is not a 'special quadratic', a quadratic such that knowledge of any of its roots enables factorization. It was suggested in [14] overcoming this problem by sending K quadratics in step 3 from which B chooses K-1 at random and asks A to send their roots first and then the protocol goes on with the remaining quadratic.

3. SUBSCRIPTION TO A PUBLIC-KEY

The problem is: A has a public-key E =(n,e), based on n=p q (RSA [15], Rabin [12] or Blum-Goldwasser [3]). A wants B to subscribe to the key, namely to get the decryption key D=(n,d). To solve this problem without compromising the key, A and B use the following cryptotransaction:

Protocol 3 - SUBSCRIPTION TO A KEY

step 1 : A publicizes E.

step 2 :

a. B chooses K random numbers (say K=10): $x_1, \ldots, x_{10}$.

b. B checks: if $\gcd(x_i, n)$ is not trivial for some x_i, then STOP. (B got the key, the chance for this is virtually zero.)

Otherwise B computes $z_i = x_i^2 \pmod n$, i=1,..10. B-->A : "z_i, i=1,..,10".

step 3 :

a. A, knowing the factorization of n, computes the 4 square roots of z_i = $\{ x_i, -x_i, y_i, -y_i \}$, i=1,..,10.

b. A uses procedure SELECT to choose one of the roots, and calls it s_i (the SELECT process makes sure that if z_i is sent twice then the same s_i is chosen).

c. A-->B : "s_i, i=1,..,10".

end {protocol 3}

Theorem 4: The protocol "Subscription to a Public-Key" ensures:

1. B gets the decryption key with probability at least $1-(1/2^{10})$.
2. An eavesdropper cannot get information from the protocol which helps him factor n.

The above protocol has several applications to cryptosystems (e.g. distribution of a group key).

4. ABSTRACTION OF THE "MENTAL POKER GAME"

4.1. SPECIFICATION OF THE GAME

For A and B to play a fair "Mental Poker Game" we need the following protocols:

1. *A protocol for Dealing Cards.* The security and verifiability specifications contain some antagonistic requirements which make the problem interesting.
2. *Protocols for other game steps:* These include discarding cards from one's hand, opening a card, etc. in a secure and checkable way.
3. *A Protocol for the Game Management* which links all the game steps together into a complete game.

4.2. DEFINITION OF CARD SETS

We define sets which are changed dynamically during the game.

ALL - the set of all the cards which is the universal (ordered) set: {1,2,3,...,52}.

AHAND (BHAND) - cards which are currently in the hand of A (B).

AUSED (BUSED) - cards which were thrown from the hand of A (B).

$ASAW = AHAND \cup AUSED$, $BSAW = BHAND \cup BUSED$,

$SAW = ASAW \cup BSAW$.

AOPEN (*BOPEN*) { *TOPEN* } - cards opened by A (by B) {to the table}.

$OPEN = AOPEN \cup BOPEN \cup TOPEN$.

$DECK = ALL$ - $\{SAW \cup OPEN\}$ - the cards currently in the deck.

$DECK_A = DECK \cup BSAW$ - cards that according to. A's partial knowledge can still be in the deck.

$DECK_B = DECK \cup ASAW$ - possible deck according to B's partial knowledge.

4.3. REPRESENTATION OF THE GAME

The game is fully represented by the card sets, so we can look at the game as a *Knowledge Set Transition System*: The Interpretation of the game as a formal system helps us to design it and to prove its properties, using formal inference about user knowledge and card sets.

States : States are positional vectors of sets which are subsets of *ALL*.

A game-state : GS = $(DECK, AHAND, BHAND, ASAW, BSAW, OPEN)$.

A special state is the illegal state which is a dead state.

The initial state is $(ALL, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

Knowledge : The player's partial knowledge of the game, at any moment of the game is also represented by a set vector. The set notation is augmented by the following: 1. *?* - an unknown set. 2. $?_i$ - an unknown set of size i, where the size is the only knowledge about it. A's Partial-Knowledge (PK) of the game is:

- PK(A) = $(?_{|DECK|}, AHAND, ?_{|BHAND|}, ASAW, ?_{|BSAW|}, OPEN)$

Transitions : The transitions are the game steps {Dealing, Discarding, Opening, Opening from *DECK*}. Any illegal game step leads to the illegal state.

Our proof technique uses assertions on knowledge and card sets, showing for example that the following are game-invariant: $ASAW \cap BSAW = \emptyset$, $DECK \cap SAW = \emptyset$,
$ASAW \cap BSAW = \emptyset$, $DECK_A \cap DECK_B = DECK$ and the fact that combining both players PK's gives the game state.

5. An Algorithm for Dealing Cards

5.1. FIRST APPROXIMATION OF THE DEALING PROTOCOL

The general idea: When B draws cards from *DECK*, they are actually offered to him by A as follows:

A knows $DECK_A = (DECK \cup BSAW) = ALL$ - $\{ASAW \cup OPEN\}$.
Using this knowledge, A tries to transfer cards he has not yet seen without revealing which cards he is offering and without being able to know which cards are chosen by B. During the process of dealing B gets a card $M \in DECK$, this card is a random card from the deck. When B gets the j cards he needs $M_1,\ldots\ldots,M_j$ he is responsible for halting the dealing without trying to look at other cards in *DECK*. Then B updates : $BHAND := BHAND \cup \{M_1,\ldots\ldots\ldots,M_j\}$;
$BSAW := BSAW \cup BHAND$

Protocol 4: THE DEALING OF CARDS PROTOCOL

- *step 1 :*
  a. A chooses $M_1,\ldots\ldots,M_{52}$ random 100-digit numbers to represent *ALL*
  b. A computes $f(ALL)=f(M_1),\ldots,f(M_{52})$, f- is A's one way function.
  c. A-->B : " f(*ALL*) ".

- *step 2 :* A tries to transfer cards from the *l* cards of $DECK_A$:
  a. A embeds the cards in $DECK_A$.
  b. Permutation choosing - (this sub-step is eliminated later, we need it just for the first approximation):
  - A chooses a random permutation of $\{1,\ldots.l\}$: P_A, and hides it unambiguously in EM(..P_A..)
  - A-->B:" EM(..P_A..) "
  - B chooses a random permutation P_B, B-->A : " P_B ".
  - A computes $P = (P_B{\cdot}P_A)$. Let $\{1,..,l\}$ be the order of $DECK_A$ derived from the order of *ALL*. P is a random permutation of it, and B does not know what P is.

- *step 3 :* A sends cards to B:
  a. A-->B : " $EM(M_{P(i)})$, i= 1, *l* "
  b. Oblivious transfers :
  (The goal of this step is to let B factor the embeddings of cards. The permutation P, the probability of success of a single transfer and the merging of the transfers of the different cards, randomizes which cards are to be factored. This is Blum's idea for sending certified mail [4].)

```
begin loop :
  for j= 1 to k { k is the size of each embedding }
  for i= 1 to l do :
    A single OBLIVIOUS TRANSFER with RECEIPT
  to enable factorization of $n_{(p(i),j)}$
end loop.
```

  c. *Getting a card :* During the above transfer process B factors all EM(M) so he gets M, then he computes f(M) and he knows which card M represents. If M ∈ *BSAW* nothing happened, else B adds M to his hand.

- *step 4 :* After B gets the number of cards he needs, he halts the protocol by
  B-->A :" stop, I got j cards ".
  end {protocol 4}

The Remaining Problems:

1. It is possible that in step 4 player B must halt the process right after he took the last card he needed, and if he does not halt, he may see an extra card from *DECK* which he is not supposed to see. We will show a solution for this 'Halting

Problem'.
2. A knows a priori the order in which the cards are offered. Even if we let B choose which n_i he wants to try to factor in any order, there is still a bias. A knows at any moment (including the end of the dealing) the current probability with which any card is given, and different cards have different probabilities at least 1-($1/l$) of the time. The solution to problem 1 will solve problem 2 as well.

5.2. SECOND APPROXIMATION OF THE DEALING PROBLEM - THE SOLUTION TO THE 'PROTOCOL HALTING PROBLEM': "THE SECRET BLOCKING"

The purpose of this approximation is to explain the idea of 'The Secret Blocking'.

The Solution to the Problems:

- 1. A (symmetrically B) has a set of public keys :
$KEY_A = REALKEY_A \cup DUMMYKEY_A$
$REALKEY_A$ is the set of keys for which A has both the encryption and the private decryption keys. For the dummy keys, A has only encryption keys and he can not decode messages encrypted by them. Half of the keys are real; half of them are dummy. (symmetrically KEY_B has the same subsets.) We assume ***temporarily*** that *KEYS* are given to the parties before the game by a ***Judge*** who knows which keys are real and which keys are dummies.
A (B) publishes all his encryption keys in a random order. B (A) can not know which keys are real, and which are dummies. We call these keys *'root-transfer keys'*.

- 2. the oblivious transfers in step 3 of protocol 4 are as follows:
a. Before A obliviously transfers a root, B-->A : " use my root-transfer key K_i " and then
A-->B : " a root s_i encrypted by this key ".
b. At first B chooses a key at random from KEY_B. If he chose a real key he may get the factorization with probability = 1/2, but if he chose a dummy key he gets information he can not decrypt. Hence the probability that B gets the factorization is 1/4.
c. It is agreed that the halting of the protocol is at the end of the loop in step 3. After B gets all the cards he needs, he must continue the transfers until the end of the loop. For M_i which he has not yet recovered, he chooses a random number from EM(M_i) not yet factored, and for its root transfer he chooses a ***random dummy-key***. Doing so he ensures that he gets information he can not decode and still he can not tell what EM(M_i) hides. Because of the random choice of root-transfer keys during the whole process A has no idea which information was blocked like this by B. This is *"The Secret Blocking"*. The secret blocking also solves the second problem. What is actually done is : B

chooses a priori which embeddings of cards to try to take and which to block. Thus P (the random permutation of the offered cards) can be chosen by A alone.

- 3. We have a *Protocol for Open Replay of the Dealing* which is used for verification. When the game is over the *Judge* uses the receipts to replay the dealing and verifies that:
  a. B got exactly the number of cards he needed, he got them from *DECK*, and he did not see any extra cards.
  b. A always used the encryption key E which B asked him to use, and did not try to check a key's status by sending some other random message.

5.3. HOW TO ELIMINATE THE CENTRAL *JUDGE*: THE SOLUTION TO THE DEALING PROBLEM

The Idea is as follows:

1. A chooses KEY_B for B.
2. A publishes the chosen encryption keys.
3. KEY_B are distributed to B using a variant of "the subscription to a public-key protocol", using one root and receipts. As a result B gets a real-key (dummy-key) with probability $= 1/2$ $(1/2)$. B takes keys until he has as many as he needs (say 30) of each subset of keys.
4. Symmetrically B chooses keys for A.
5. The fact that B (A) knows the encryption and the decryption keys of all keys in KEY_A (KEY_B), does not compromise the secret blocking.

The Improvements to The Dealing Protocol are :

1. In step 2.b the permutation (P) is chosen at random by A alone.
2. In step 3.b B randomly chooses which *card embeddings* to try to take. He applies the secret blocking to embeddings he decides not to take.
3. In step 3.b B halts the dealing and moves to step 4 at the end of the loop.

As a result of the improvements the following theorem holds:

Theorem 5: The "dealing of cards protocol" is correct according to its specification:
a. If no player cheats, then when a player draws the cards, the following properties hold: *Fairness*, *Disjointness of Drawn Cards*, *Security*, *Verifiability*.
b. Any case of cheating is detectable.

6. THE COMPLETE GAME OF TWO PLAYER MENTAL-POKER

We design a main protocol called *The Game Manager* which organizes the game and links the different steps and we design *Protocols for other game steps* as well. The game steps are:

1. Discarding a card : A moves a card M from *AHAND* to *AUSED* in a secure and checkable way.

2. Opening a card : A moves a card M from *AHAND* to *AOPEN* in a checkable way.

3. Opening a card from *DECK*: first, using protocol 4, A gives a card M to B then B opens M.

It is easy to design these protocols since at the beginning of each of them a new random code of the abstract card sets is used. The order of *ALL* is the interface between steps and we can prove the following theorem:

> **Theorem 6:** The two player mental poker game is fair, secure, checkable and a direct simulation of the regular game (using cards) as was specified.

7. GENERALIZATION: THE MULTI-PLAYER MENTAL POKER

7.1. THE PROBLEM IN MULTI-PLAYER GAME

In the two player game the cards are offered to B from the set $DECK_A =$ $DECK \cup BSAW$, and B adds the opened cards he did not previously see to his hand. The disjointness of the cards he takes and cards that have already been seen by A (at any given moment) is a consequence of the fact that the combining both players partial knowledge is the full knowledge of the game, and that *DECK*, *ASAW* and *BSAW* are mutually disjoint while their union is *ALL*. How can we guarantee, however, that B takes cards only from *DECK* and does not get any additional partial information, while $DECKA_1^{'} \cap DECKB \not\models DECK$ in the generalized situation? We must somehow let all the players participate in the dealing and still keep the mutual privacy and security constraints.

Assumption : All messages are sent to all players. This is a minimal assumption, because otherwise if even two out of the K players can communicate privately, they can make a coalition and get an advantage over the others just by knowing each others' hands. Also, in order to be able to replay the protocols, we assume that every message is acknowledged by all the players.

The Changes : For each player j we define the following sets: $HAND_j$, $USED_j$ and SAW_j are respectively the cards in his hand, cards he already used and their union. During the game we keep

$\{ SAW_j \cap SAW_i = \emptyset$ for i $\not=$j $\}$ and $\{ SAW_j \cap DECK = \emptyset \}$.

Suppose there are K players. Let B be the K-th player and A_i, i=1,..,K-1 all the others. We define

$DECKTOB =_{j=1}^{k-1} \cap (DECK_j) = DECK \cup BSAW$. This is the set that player B,

who is taking the cards, is allowed to see and to choose cards from. We call these cards Candidate Cards, because such a card is a candidate to be drawn by B, namely if a card $M \in$ *DECKTOB* - *BSAW* then $M \in$ *DECK*, and B can take it.
The multi-person dealing protocol is a two-stage protocol: *DECKTOB* generation stage and Card drawing stage. These two stages are two coroutines, each of them has a ***current state*** and they run concurrently.

7.2. THE SOLUTION : A PROTOCOL FOR MULTI-PLAYER DEALING OF CARDS

Protocol 5 : MULTI-PLAYER DEALING
(A_i, i=1,.,K-1 deals cards to $B=A_k$. They start at stage A.)
current state of stage A is : "begin the stage in step 1".
current state of stage B is : "begin the stage in step 6".

Stage A: DECKTOB GENERATION
The stage starts at its current state:
step 1 : The K-1 players choose a common random permutation of 1,.....,52 : Q (B does not know what Q is). They embed Q unambiguously in EM(Q), and transfer it to B. (The communication between the k-1 players can be done using a group key, see section 3.)
step 2 : Every player A_i chooses his own ***private current code*** of *ALL* : ALL^i
each A_i-->B : " $Q(f_i(ALL^i))$ ".
(The players do not reveal the cards in the right order, but rather, a random permutation of them.)
step 3 : For player A_i let : $DECK_i = DECK \cup (\cup_{j:j\neq i}\{SAW_j\}) = ALL - SAW_i$. A_i embeds each card $M \in DECK_i$ in EM(M).
step 4 : Each A_i chooses a private random permutation P_i.
A_i-->B : " $P_i(EM(\ DECK_i))$ "
(The cards are offered in order P_i, A_i has to remember this order.)
step 5 : Opening of cards : B tries to open cards using OBLIVIOUS TRANSFERs, alternately with A_i, i=1,., K-1. He makes iterations over the embedded cards as in the two player case. During this process B can get the following information :
a. *Factoring of a card embedding :* B gets a card code M of some of the other players A_i. He can compute $f_i(M)$, but this gives him no idea what M is because he gets only the place of M in the permutation Q which is a random permutation and M is just a random number.
b. *Getting a candidate card :* During the transfers B realizes that a card is offered to him by *all* the other players, (the same place in Q was revealed in all $Q(ALL^i)$). This card is either in *DECK* or in *BSAW* so it belongs to *DECKTOB* (it is a candidate card). Suppose B needs v cards and during the process he gets v random candidates. Then the players remember ***the current state of stage A*** and go to Stage B.
end {**Stage A**}

Stage B : CARD TAKING
(In this stage only B and one of the players (A_1) are playing, but the others get and acknowledge all messages.)
The stage starts at its current state:
step 6 : A_1 chooses a new code of *ALL*: ALL^1.
A_1-->B " $f_1(ALL^1)$ " (This time without permuting the order.)
step 7 : A embeds each $M \in DECK_1$ in EM(M).
A_1-->B " $P_1(EM(DECK_1))$ ". (He uses the *same* permutation P_1 he used in stage A; the permutation of $DECK_1$ is the ***interface*** between the two stages.)
step 8 : B and A_1 use iterations of OBLIVIOUS TRANSFERs in order to let B factor the embeddings. B knows which card in the permutation P_1 is a candidate, using the SECRET BLOCKING he chooses to open only candidates.
Taking a card : When B gets all the factors of the embedding of a candidate card, he recovers M and computes $f_1(M)$. If $M \notin BSAW$ he takes it. If B gets all the cards he needs, he stops the process by B-->A_1: "stop, I got j cards". If he has already seen some of the candidates, then the players return to stage **A**.
end {**Stage B**}
end {**protocol 5**}

The reduction of the multi-player case to several two-player protocols implies the following:

> **Theorem 7:** The protocol for Multi-player Dealing of Cards simulates dealing of cards and has the specified properties of security, verifiability and fairness.

8. CONCLUSIONS

We presented cryptoprotocols which can be used with a public-key cryptosystem. The subscription to a public-key and the secret blocking protocols are cryptographic tools, augmenting the power of public-key systems. Developing these tools and their applications and solving the multi-player mental poker game extends our knowledge of the power of cryptographic techniques, the range of applications of these techniques, and the boundaries between the possible applications and the impossible ones. The study of these subjects is one of the main targets of recent cryptographic research.

In designing the protocols, we used a methodology that can be used for designing and proving the correctness of long cryptoprotocols. We observe four main design stages:
1. The axiomatic stage: We have two kinds of axioms: a. The underlying mathematics. b. The computational environment: rules of communication, user behavior, etc.
2. The basic Cryptographic Techniques: Based on our axioms, we use or construct basic algorithms and cryptotransactions like the RSA system, Oblivious Transfer,

Number-Embedding.
3. Top-Down Design of the protocol: The problem at hand is divided into sub-problems (an analogous to modular design of a computer program). For every sub-problem we develop a cryptoprotocol using the basic tools of stage 2. We take care of the security and other specified properties of the sub-problems' protocols, at the same time ensuring the specified properties of the whole process. We use an inference system which includes "security logic" and "process logic". (In our case we prove formal assertions about card sets and users' information and we use the global order of the cards to concatenate steps.)
4. The Process Protocol: After stage 3 the process is executed over the communication channels according to the rules of the original process. We also handle additional administrative communication which we ignored when we concentrated on the problem.

This approach of divide and conquer (using the same or other system axioms and proof techniques) will undoubtedly be used in other complex and long cryptoprotocols that will be designed in the future.

Acknowledgements

I wish to thank my teacher Zvi Galil for his help and encouragement, and Bruce Abramson, Bruce Hillyer, Brian Kwartler and Carmi Weinzwig for their useful comments.

References

1. Angluin D. Lecture notes on the complexity of some problems in number theory. Tech. Rept. 243, Dep. of Computer Science, Yale University, August, 1982.

2. Berlekamp E.R. "Factoring Polynomials over Large Finite Fields." *Mathematics of Computation 24* (July 1970), 713-735.

3. Blum M. and S. Goldwasser. An Efficient Probabilistic Public-Key Scheme Which Hides All Partial Information. to appear in Proceedings of Crypto84, 1984.

4. Blum M. Three Application of the Oblivious Transfer. University of California at Berkely, September, 1981.

5. Blum M. Mental Poker. University of California at Berkely, 1982. to appear

6. Blum M. "How to Exchange (Secret) Keys." *ACM Transactions on Computer Systems 1*, 2 (May 1983), 175-193.

7. Diffie W., and M.E. Hellman. "New Directions in Cryptography." *IEEE Transactions of Information Theory IT-22* (November 1976), 644-654.

8. Goldwasser, S. and Micali S. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. Proceedings of the 14 Annual ACM Symp. on Theory of Computing, ACM-SIGACT, May, 1982, pp. 365-377.

9. Knuth D. E.. *The Art of Computer Programming.* Volume 2: *Seminumerical Algorithms.* Addison-Wesly, Reading, Massachusetts, 1981.

10. Lipton R. How to Cheat at Mental Poker. Proceeding of the AMS short course on Cryptography, AMS, January, 1981.

11. Niven I. and Zuckerman H.S.. *An Introduction to the Theory of Numbers.* Wiley, New York, 1981.

12. Rabin M. Digitalized Signatures and Public-key Functions as Intractable as Factorization. Tech. Rept. LCS/TR-212, MIT, January", 1979.

13. Rabin M. Probabilistic Algorithms in Finite Fields. Tech. Rept. LCS/TR-213, MIT, January, 1979

14. Rackoff C., S. Micali and M. Fischer. A Secure Protocol for the Oblivious Transfer. Eurocrypt 84, La Sorbonne, Paris, April, 1984.

15. Rivest R. , Shamir A., Adleman L. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Communications of the ACM 21*, 2 (February 1978), 120-126.

16. Shamir A. "How to Share a Secret." *Communication of the ACM 22*, 11 (November 1979), 612-613.

17. Shamir A., Rivest R. Adleman L. Mental Poker. In *Mathematical Gardner*, Klarner D. E., Ed.,Wadsworth Intrntl, 1981, pp. 37-43.

18. Solovay R., and Strassen V. "A Fast Monte-Carlo Test of Primality." *SIAM Journal on Computing 6* (March 1977), 84-85.

19. Yung M. K-Player Mental Poker. Master Th., Tel-Aviv University, March 1982.