POKER PROTOCOLS

Steven Fortune
Michael Merritt

AT&T Bell Laboratories
600 Mountain Ave
Murray Hill, NJ 07974

1. INTRODUCTION

The situation is quite serious. After four years of research, there has been no satisfactory way for a group of card sharks to play poker over the phone. Until now. In this paper, we present a new method for playing 'mental poker,' discuss its significance, and mention some of the further questions it raises. Ante up.

The rules for mental poker are just like regular poker, except that players communicate over the phone, and there are no physical cards. The hard part of mental poker is dealing the cards. Hands must be random and disjoint, and players should not be able to claim to have any cards but those dealt (a sleeve will hold as many 'virtual cards' as angels will fit on the head of a pin).

Playing mental poker is a difficult problem for a number of reasons. The foremost reason is that it is impossible, a result due to Shamir, Rivest and Adleman.[1] Of course, this is an information-theoretic result, and the same reference presents a method for playing mental poker that relies on the difficulty of inverting certain cryptographic transformations. Unfortunately, a cryptographic flaw allows players to determine the color of each other's cards.[2] This set the stage for a new implementation devised by Goldwasser and Micali, which was proven to hide all partial information (up to an explicit cryptographic assumption).[3] Unfortunately, this implementation works only for two players, which is a very restricted kind of poker. Next, Barany and Furedi devised a protocol that permits three or more players to play poker,[4] but only if players are not permitted to form coalitions. If two players conspire, they can learn the contents of everyone else's hands. The following section discusses this history of mental poker in more detail, outlining the key ideas, contributions and limitations of this earlier work.

This paper presents a new way of playing mental poker. Unlike earlier solutions, it is secure against coalitions, permits any number of players, and uses inexpensive, highly secure cryptographic techniques. The protocol does require the participation of a trusted party to shuffle the cards. However, thereafter the trusted party does not participate in the protocol. The protocol can be easily adapted to play almost

G.R. Blakley and D. Chaum (Eds.): Advances in Cryptology - CRYPTO '84, LNCS 196, pp. 454-464, 1985.

all types of poker known to the authors.

Of course, poker is a metaphor for any system in which users should have only partial information about the dynamic allocation of resources. Beyond this, the poker protocol presented here takes on a broader significance because of the simple tools used in its implementation.

2. THE HISTORY OF MENTAL POKER

2.1 A Protocol based on Commutativity

To our knowledge, the first attempt to play mental poker was made by Niels Bohr during a skiing vacation near Oberaudorf in 1933. At Bohr's instigation, he, his son Christian, Felix Bloch, Carl Friedrich, and Werner Heisenberg attempted to play poker without cards, each player calling out the contents of his nonexistent hand. Heisenberg reports "The attempt was made, but did not lead to a successful game." Apparently, the players did not pursue the problem further, finding adequate challenge in other research areas.[5]

Much later, the problem was independently posed by Robert Floyd. This led to the first formal results on mental poker, in a 1979 research report by Adi Shamir, Ronald Rivest, and Leonard Adleman, later published in *The Mathematical Gardner*[1]. In the spirit of modern cryptography, they began by proving the problem impossible to solve. The argument is very simple: if Alice and Bob are playing poker, either Bob can claim to have a straight-flush every time, or Alice has enough information to determine Bob's hand. Having established the problem as adequately challenging, Shamir, *et al* proceeded to solve it. Their solution circumvents the impossibility result by exploiting 'hidden' information. Alice knows Bob's hand in the sense that it is the unique solution to a computational problem, but finding that solution is beyond her capabilties. Verifying the solution, once Bob divulges his hand, is easy.

More specifically, this first poker protocol utilizes the power of *commutative* cryptosystems. Let E_A and D_A be Alice's encryption and decryption functions, respectively, and let E_B and D_B be Bob's. Suppose that $E_A(D_B(x))=D_B(E_A(x))$ and $E_B(D_A(x))=D_A(E_B(x))$ for all messages x. Then Alice and Bob play a hand of poker as follows (it is a simple exercise to extend this protocol to three or more players).

Let the deck of cards be any encoding of the set $\{1,\ldots,52\}$ appropriate for the cryptosystem. Alice encrypts each card in the deck separately, randomly orders the resulting set $\{E_A(1),\ldots,E_A(52)\}$, and sends it to Bob.

Bob chooses five encrypted cards at random, say $\{E_A(18), E_A(24), E_A(27), E_A(31), E_A(39)\}$, and sends them back to Alice, who now knows her hand is $\{18,24,27,31,39\}$.

Next, Bob chooses five different encrypted cards, say $\{E_A(3), E_A(12), E_A(15), E_A(35), E_A(41)\}$, encrypts them, and sends the randomly ordered set $\{E_B E_A(3), E_B E_A(12), E_B E_A(15), E_B E_A(35), E_B E_A(41)\}$ back to Alice.

Alice decrypts each element of the set, and sends the resulting set, $\{E_B(3), E_B(12), E_B(15), E_B(35),$

$E_B(41)\}$, back to Bob.

Bob decrypts the set to get his hand, {3,12,15,35,41}.

Once the hand has been played, Alice and Bob exchange their encryption keys and verify that each played fairly (until this point, both players could claim to have any hand they like, so long as it did not happen to intersect their opponent's hand). Both Alice and Bob's encryption keys must be uniquely determined by the messages sent during the deal. Otherwise, Alice or Bob could divulge the key which decodes the best hand. It is also important that the encryption function hide *all* of the message--leaking even a single bit about a card (such as its color) can make a significant difference.

Shamir, Rivest and Adleman suggest a particular commutative cryptosystem for implementing their protocol. It is based on modular exponentiation. Alice and Bob agree on a large odd prime number n, and separately choose secret keys $k=A$ or $k=B$, where $gcd(A,n-1)=gcd(B,n-1)=1$. Then $E_k(x)\equiv x^k (mod\ n)$ and $D_k(x)\equiv x^z (mod\ n)$, where $kz\equiv 1 (mod\ n-1)$.

2.2 These Cards are Marked!

Shortly after this protocol and implementation appeared in a technical report, Lipton observed that this implementation leaks at least a bit of information[2]. This is because exponentiation modulo n preserves *quadratic residues*. A number x is a quadratic residue modulo n provided $x\equiv y^2\ (mod\ n)$ for some y. Otherwise, x is a quadratic nonresidue. Half of the integers are quadratic residues modulo n, for n a large prime. For such n it is easy to determine whether a number is a quadratic residue. Finally, since k must be odd, $x^k (mod\ n)$ is a quadratic residue if and only if x is. Thus, knowing which cards are quadratic residues, and comparing these against the encrypted cards Alice sends him, Bob has about a bit of information per card.

Of course, the cards could be encoded originally so that they are all quadratic residues (or all quadratic nonresidues). Lipton discusses this and other suggestions for strengthening the cryptosystem, but notes that there is still no guarantee that the result is secure. Indeed, the indication is that bits may still leak.

2.3 A Provably Secure Two-Person Game

Some three years after Lipton's observation, Goldwasser and Micali published a new protocol for playing mental poker[3]. A major achievement of their work was a proof that discovering a single bit of an opponent's hand was equivalent to solving an apparently intractable problem (factoring, index finding or deciding quadratic residuosity with respect to composite moduli). Unfortunately, their protocol works only for two players.

The details of their protocol are beyond the scope of this survey. What follows is a very simplified description, intended to explain why it works for only two players.

Our friends Alice and Bob will play again. Alice shuffles a deck of cards, encrypts it using a function A, and sends the encrypted deck to Bob. Similiarly, Bob shuffles a deck, encrypts it using B, and sends it to Alice. Micali and Goldwasser show how Bob can ask Alice to decrypt one card of her deck, without Alice knowing which card she has decrypted. This technique is the crux of their protocol and uses some clever computational number theory; essentially Bob asks a question about every encrypted card, but for

only one card does Bob gain enough information to decrypt the card. (There is a later verification step that ensures that Bob didn't decrypt more than one card.)

So how is the game played? Suppose Bob has drawn card x from Alice's deck. He then removes $B(x)$ from his own encrypted deck, so when it is Alice's turn to draw, she can't choose $B(x)$. Similiarly, any card drawn by Alice is removed from her deck, so Bob can't draw it. Neither player knows what cards have been removed from the opponent's deck, since the decks are encrypted.

Now it is clear why the protocol works for only two players. Alice draws from Bob's deck, which does not contain Bob's hand, and Bob draws from Alice's deck, which does not contain Alice's hand. If Charles wants to play, which deck does he choose from? If he chooses from Bob's deck, he might get a card Alice has, and if he chooses from Alice's deck, he might get a card Bob has.

2.4 Three or More Players

What happens with three or more players? Suppose Charles wishes to play with Alice and Bob. To keep Alice from falsely claiming to have a straight-flush, Bob and Charles must together have enough information to determine her hand. But neither alone need have this information. And as long as they cannot share information, they remain ignorant of Alice's hand. Thus, the impossibility argument of Shamir, *et al*, does not work with three or more players.

This observation was made by Barany and Furedi[4], who also present a simple protocol for mental poker for three or more players. Let's see how Alice, Bob and Charles can use this protocol to play poker. Initially, each player chooses a random permutation of the deck, A, B and C, respectively.

Next, Bob and Charles send B and C to Alice. They do this secretly, so that they remain ignorant of each other's permutation.

Now Alice secretly sends BA^{-1} to Charles, and CA^{-1} to Bob.

Now the cards are ready to be dealt. Let's jump ahead to a point where Alice, Bob and Charles already have some cards, the sets H_A, H_B and H_C, respectively, and see how new cards will be dealt. (Initially, these are the empty sets). We assume that the hands are disjoint so far.

At this stage, each player has the following information.

Alice knows A, B, C, H_A.
Bob knows B, CA^{-1}, H_B, $A(H_A)$, $A(H_B)$, $A(H_C)$.
Charles knows C, BA^{-1}, H_C, $A(H_A)$, $A(H_B)$, $A(H_C)$.

Suppose Bob wants a new card. He gets it from Charles as follows. Charles chooses a number x not in $A(H_A)$, $A(H_B)$ or $A(H_C)$, and sends $BA^{-1}(x)$ to Bob, from which Bob computes his card, $y = A^{-1}(x)$. Bob adds y to H_B, and both Bob and Charles add x to $A(H_B)$. Charles gets a card from Bob in a similar way.

When Alice wants a new card, she gets it from either Bob or Charles, say Bob. Bob chooses a number x not in $A(H_A)$, $A(H_B)$ or $A(H_C)$, and sends x to both Alice and Charles. Alice adds $y = A^{-1}(x)$ to H_A, and Bob and Charles add x to $A(H_A)$.

Alice plays a special role in this protocol. Thinking of her permutation as a shuffled deck of cards, everyone but Alice knows which cards in this shuffled deck they each hold. For instance, Bob and Charles may know Alice has the 3^{rd} and 4^{th} cards in the deck, Bob has the 34^{th} and 51^{st} and Charles has the 22^{nd}. By picking cards in A that have not yet been dealt, Bob and Charles can keep all hands disjoint. Because only Alice knows how the cards are ordered by A, the cards Bob and Charles pick will be randomly chosen.

The assumption that players will not collude is crucial to the protocol. If any two players share their knowledge, they learn not only each other's hands, but their opponent's hands, as well. No one would bet real money under these conditions. In Section 3, we show how Alice's special role can be played by a trusted party during an initialization stage, in such a way that players who collude learn only the contents of each other's hands.

2.5 Other Poker Protocols

A protocol based on ideas similiar to the Goldwasser-Micali protocol was independently proposed by Yung.[6] Yung's protocol improves on the Goldwasser-Micali protocol by allowing more than two poker players. Unfortunately, his protocol is quite complex, and like the Barany-Furedi protocol, it is not secure against collusion: two players collectively have enough information to deduce other players' hands as well.

One approach to prevention of collusion was suggested by Fich and Goldwasser.[7] The Fich-Goldwasser protocol is based on the Barany-Furedi protocol; the novelty is that it is not possible to transmit secret information in the messages of the protocol itself. If all communication between participants is restricted to protocol messages, then collusion is impossible. Of course, if there is some secret channel between players, say a surreptitious telephone line, then the protocol suffers the same defect as the Barany-Furedi protocol, and two colluding players can learn all hands.

3. PRACTICAL MENTAL POKER

3.1 Introduction

Can we construct a poker protocol for three or more players that is secure against collusion? We now do so. Our protocol uses the "distributed-information" technique of Barany and Furedi. In addition, it uses one-way functions to authenticate information, and the services of a "Card Salesman" as a trusted participant.

The Card Salesman participates in the protocol at the beginning of play. He receives a small amount of secret information from each player, then publicizes information that allows play to proceed. The Card Salesman must be a trusted participant, for he has enough information to discover all players' hands. The advantage of a Card Salesman over a trusted dealer (in which case the poker problem is trivial) is that the Card Salesman need only participate at the beginning of play.

In many ways the role of Card Salesman is analogous to the manufacturer of a deck of cards. Serious poker players insist on beginning any poker game with a new deck of cards, in a box still sealed by some trusted manufacturer. The players then have some assurance that the cards are not marked. Both in the case of the card manufacturer and of the Card Salesman, the cost of the trusted participant is small.

The poker protocol also requires the use of one-way functions. A one-way function f is a function that is easy to compute and hard to invert. One-way functions are valuable authentication tools. For instance, computer systems often store encryptions of passwords, rather than cleartext passwords.[8] Although there is no proof that there is any easy-to-compute function that is hard to invert (since such a proof would imply $P \neq NP$), practical one-way functions are easy to construct. This is because there is no need to construct the inverse of the one-way function, quite in contrast to the case of public-key or private-key encryption methods. We assume the existence of a one-way function f that is either one-one, or if not one-one then, given $z=f(y)$, it is computationally hard to find any x so that $f(x)=z$. Further discussion of one-way functions can be found in the references[8] [9] .

The poker protocol has the following general format. At initialization, each player chooses some secret information. At later stages, each player makes a move that either is based on a random choice, or is completely determined by the secret information initially chosen. The fairness of the protocol depends upon the player abiding by the initially chosen secret information. To convince other players that he is not cheating, at the beginning of play each player broadcasts his secret information, encrypted by a one-way function. At the end of the game, each player broadcasts his secret information, unencrypted. Then all players can check that all other players followed the protocol. Note that since all messages besides secret messages are broadcast to all players, the information needed to check other players behavior is available. Also, since one-way functions are hard to invert, after play is over, a player cannot broadcast secret information different from what he was using during play.

As long as the Card Salesman and at least one player play fairly, no group of colluding players can gain an unfair advantage over other players. The proof of this assumes that the one-way function cannot be inverted. Actually, a stronger assumption is necessary, that no statistical information at all about the preimage of a function value can be inferred. This assumption becomes apparent in the analysis of the protocol, as we ignore the fact that the encrypted secret information has been published.

3.2 The protocol

The poker protocol assumes that two network services are available: the ability to send secret messages between pairs of players, and the ability to broadcast a message to all players. Actually, secret messages are only sent at the initialization of the protocol, and then only to the Card Salesman. All other messages are broadcast to all players. We assume that the network reliably provides these two services.

The duty of the Card Salesman is to choose a random permutation π that encodes players hands. Suppose Alice has a hand H_A. Of course H_A is not known to other players, but $\pi(H_A)$ will be public knowledge. No player has information about π beyond the value of π on his hand. To draw a new card, the player must choose some $y=\pi(x)$ not in any other player's hand. This is possible since the player knows the π-encoded form of the other players' hands. The poker protocol then reveals to the player $x=\pi^{-1}(y)$ without letting any other player know x.

So how does a player, say Charles, draw a card? Before the game starts, Alice, Bob, and Charles each choose a random permutation α, β, and γ, respectively, and transmit their permutation secretly to the Card Salesman. The Card Salesman computes the product $\Delta=\alpha^{-1}\beta^{-1}\gamma^{-1}\pi^{-1}$ and broadcasts it to all players. Now suppose Charles wishes to draw a card. He randomly chooses some $y=\pi(x)$ not in any other player's hand, and broadcasts y and $\Delta(y)$. Alice now computes and broadcasts $\alpha(\Delta(y))$. Bob computes and broadcasts $\beta(\alpha(\Delta(y)))$. Finally, Charles computes $\gamma(\beta(\alpha(\Delta(y))))=x$, and of course does not broadcast it. Note that the same permutations α, β, γ can be used the next time Charles draws a card. However, permutations α, β, γ cannot be used to draw cards for a different player.

We now describe the complete poker protocol for three players, Alice, Bob, and Charles. The generalization to more than three players is straightforward.

Initialization:

1. The Card Salesman randomly chooses a permutation π.
2. Alice chooses three permutations α_A, α_B, α_C. Similiarly, Bob and Charles each choose three permutations β_A, β_B, β_C and γ_A, γ_B, γ_C. All permutations are transmitted secretly to the Card Salesman, and their encryptions using the one-way function are broadcast.
3. The Card Salesman computes and broadcasts the products $\Delta_A=\beta_A^{-1}\gamma_A^{-1}\alpha_A^{-1}\pi^{-1}$, $\Delta_B=\gamma_B^{-1}\alpha_B^{-1}\beta_B^{-1}\pi^{-1}$, and $\Delta_C=\alpha_C^{-1}\beta_C^{-1}\gamma_C^{-1}\pi^{-1}$.

For Charles to draw a card:

1. Charles randomly chooses $y=\pi(x)$ not in any player's hand and broadcasts y and $\Delta_C(y)$.
2. Alice broadcasts $\alpha_C(\Delta_C(y))$.
3. Bob broadcasts $\beta_C(\alpha_C(\Delta_C(y)))$
4. Charles computes $x=\gamma_C(\beta_C(\alpha_C(\Delta_C(y))))$.
5. All players record that Charles has $y=\pi(x)$ in his hand.

For Alice to draw a card, Alice publishes y and $\Delta_A(y)$, then Bob and Charles broadcast $\beta_A(\Delta_A(y))$ and $\gamma_A(\beta_A(\Delta_A(y)))$, respectively. Similiarly, for Bob to draw a card, Bob publishes y and $\Delta_B(y)$, then Charles and Alice broadcast $\gamma_B(\Delta_B(y))$ and $\alpha_B(\gamma_B(\Delta_B(y)))$, respectively.

End of play: Each player publishes his permutations, and checks that every other player played fairly. Then the debts are settled.

3.3 Correctness

Is it possible to play poker with this protocol? Certainly. It is clear that one round of the protocol allows a player to draw a card not in anyone else's hand. The more important question is: is anyone willing to play poker with this protocol? I.e., is it possible to guarantee absence of cheating? It is clear that before any cards are dealt, and in the absence of collusions among the players, π is random, so a player has no better strategy for drawing cards than random choice. But what happens in the presence of collusions, and after cards are dealt? We show that as long as the Card Salesman and at least one player play fairly, that is, they choose their permutations randomly with uniform distribution and conceal them from other players, then no other player or group of colluding players can gain any information on cards

not in their own hands.

How can we analyze the protocol? We wish to determine the probability distribution of π, given the information a player or group of players have at some point during play. We know that originally π was chosen randomly by the Card Salesman. However, the Card Salesman publishes the permutations Δ, and as play proceeds information is made public about each player's private permutations. To analyze the probability distribution of π, even in the presence of these additional constraints, we set up a state space that captures the possible values for π and also all of the permutations initially chosen by the players. Note that the state space does not capture the later random choices made by the players. The state space has a probability distribution, given by the probabilities with which the players choose their permutation. A player's knowledge is modelled as a subset of the possible states, specifically, all the states satisfying the player's knowledge. We determine the conditional probability distribution for π, given that the state must satisfy the knowledge of a player or group of players.

The analysis is given in terms of three players Alice, Bob, and Charles, assuming that Alice plays fairly, and Bob and Charles possibly collude. The generalization to more than three players is straightforward.

The state space has variables A_A, A_B, $\cdots$, Γ_C and Π. Each of these variables has as possible values a permutation on $1 \cdots 52$. A state is a particular set of values for the variables, $A_A=\alpha_A$, $A_B=\alpha_B$, $\cdots$, $\Gamma_C=\gamma_C$, and $\Pi=\pi$. We use upper case letters for variables and lower case letters for particular values. Knowledge is denoted by a set of equations involving the variables; the set of equations specifies all states satisfying the equations.

What is known, publicly and privately? We summarize it as follows.
Alice's knowledge, K_A: $A_A=\alpha_A$, $A_B=\alpha_B$, and $A_C=\alpha_C$
Bob's knowledge, K_B: $B_A=\beta_A$, $B_B=\beta_B$, and $B_C=\beta_C$
Charles's knowledge, K_C: $\Gamma_A=\gamma_A$, $\Gamma_B=\gamma_B$, and $\Gamma_C=\gamma_C$
Public Knowledge, P (consisting of all messages that have been broadcast):

$\Delta_A=B_A^{-1}\Gamma_A^{-1}A_A^{-1}\Pi^{-1}$

$\Delta_B=\Gamma_B^{-1}A_B^{-1}B_B^{-1}\Pi^{-1}$

$\Delta_C=A_C^{-1}B_C^{-1}\Gamma_C^{-1}\Pi^{-1}$.

For each x in $\Pi(H_A)$: $B_A(\Delta_A(x))=\beta_A(\Delta_A(x))$ and $\Gamma_A(\beta_A(\Delta_A(x)))=\gamma_A(\beta_A(\Delta_A(x)))$

For each x in $\Pi(H_B)$: $\Gamma_B(\Delta_B(x))=\gamma_B(\Delta_B(x))$ and $A_B(\gamma_B(\Delta_B(x)))=\alpha_B(\gamma_B(\Delta_B(x)))$

For each x in $\Pi(H_C)$: $A_C(\Delta_C(x))=\alpha_C(\Delta_C(x))$ and $B_C(\alpha_C(\Delta_C(x)))=\beta_C(\alpha_C(\Delta_C(x)))$

Lemma. Let $k=|H_B \bigcup H_C|$. In any probability distribution where Π, A_A, A_B, and A_C are each uniformly distributed and independent of all other variables, then for any x not in $H_B \bigcup H_C$, y not in $\Pi(H_B) \bigcup \Pi(H_C)$, $\Pr[\Pi(x)=y \mid K_B K_C P]=1/(52-k)$.

Proof: We need to analyze the set of states consistent with the partial information K_B, K_C, and P; in particular, we need to know the possible values of Π. Say that π behaves correctly if for all x in H_B, $\pi(x)=\Delta_B^{-1}(c_B^{-1}(a_B^{-1}(b_B^{-1}(x))))$ and for all x in H_C, $\pi(x)=\Delta_C^{-1}(a_C^{-1}(b_C^{-1}(c_C^{-1}(x))))$ (that is, π has the correct value on H_B and H_C). Note that all these values are part of the partial information K_B, K_C, and P. We claim that if π behaves correctly, then there is a state with $\Pi=\pi$ satisfying the equations in K_A, K_B, K_C, and P, and if π does not behave correctly, then there is no state with $\Pi=\pi$ satisfying the equations.

Furthermore, once π is chosen, since the variables B_A, B_B, B_C, Γ_A, Γ_B, Γ_C all have values fixed at β_A, β_B, β_C, γ_A, γ_B, γ_C, respectively, the values of A_A, A_B, and A_C are uniquely determined. This follows by just examining the equations. Since each of Π, A_A, A_B, and A_C are uniformly distributed and independent, each state arising from a choice of π is equally likely. What proportion of the choices of π satisfy $\pi(x)=y$? Clearly, $1/(52-k)$.

QED

So can Bob and Charles collude? They can of course tell each other their own cards. But by the Lemma, they can gain no information on cards that are not in their own hands. Thus, they cannot infer the cards in Alice's hand, nor can they do better than a random choice in choosing a card to draw. Of course, this assumes that Alice plays fairly. But it is always in Alice's interest to play fairly, since she is then protected from cheating by other players.

As a technical note, we observe that the Card Salesman could choose the players' private permutations in addition to the permutation π. Then the Card Salesman would broadcast the permutations Δ, as before, and secretly communicate to each player his private permutations. It would be possible to implement this protocol, but it is probably easier to have all secret communication directed to the Card Salesman, as in the protocol presented.

3.4 The Many Flavors of Poker

While purists complain, the name poker applies to a wide variety of sins. Simple games like Seven-card Stud or Five-card Draw can be played using the protocol above directly. Cards can be drawn from the deck, discarded and turned face-up. More complex games are easily accommodated, but there are a few we cannot play.

One author plays an abomination called Indian, in which each player is dealt a single card, face-down. Everyone's card is then shown to every other player, but remains unknown to its holder. After a round of betting, with opportunities to fold, the high card wins. A simple adaptation of our protocol allows one to play this game, and in general to identify a card to any subset of players in a way that can be verified later.

Other games involve passing cards from one player to another. This is easy to do once, just by passing the encrypted card, which is then decoded for the new recipient. But in the game Anaconda, cards are passed more than once, and this implementation allows one to determine that a card one passed earlier is (or is not) being passed in subsequent rounds. A secret and subsequently authenticable message could be used to pass a card and avoid this problem, but the game is hardly worth the added effort.

One thing we cannot do is return one or more cards to the deck and reshuffle it. We don't know any poker games which require this, but they probably exist. In fact, we need to buy a new deck from the Card Salesman for every hand of poker we play. Luckily, decks are cheap, and it is easy to buy lots of them during a single initialization.

4. DISCUSSION

The protocol presented here is practical enough to implement and run in real time, even on networks of

small home computers. With the proper choice of one-way function (and Card Salesman), it is actually secure enough to use. This alone is a great improvement over previous solutions.

As mentioned previously, the protocol is also significant because the tools used to play poker are very simple--products of permutations to hide information, one-way functions to permit verification. This point raises several interesting questions. What other, more powerful capabilities can be implemented using such simple tools? A notable example is the probabilistic signature scheme due to Rabin,[9] which uses only one-way functions. Is there a way of characterizing the power of these tools and of establishing their limitations?

The poker problem is trivial if there is a trusted Dealer to distribute cards. The Card Salesman is not nearly as expensive as a Dealer (we claim), but we'd really like to get rid of him completely. The protocol here requires the Card Salesman to choose the random Π and to compute and broadcast the permutations Δ. Thus we would like to find a way of broadcasting a product of n secret permutations without divulging any factors. The RSA, and other commutative encryption schemes could be used to do this. Can it be done with only one-way functions and/or private-key systems?

Poker differs from some applications, in that authentication can be carried out in a single stage, after the remainder of the protocol. Other protocols, such as the secret-ballot elections of Chaum[10] and Merritt,[11] or electronic funds transfers, require authentication during the protocol itself. How important is this distinction, and what tools are required to implement the two types?

4.1 Acknowledgements

We would like to thank Barbara Smith-Thomas and Joan Feigenbaum for their helpful comments on this paper.

References

1. Shamir, A., Rivest, R., Adleman, L., "Mental Poker", in *Mathematical Gardner*, D.E. Klarner, ed., Wadsworth International, 1981, pp. 37-43.
2. Lipton, R. "How to Cheat at Mental Poker," Proceedings of the AMS Short Course in Cryptography, 1981.
3. Goldwasser, S., Micali, S., "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information," Proceedings of the 14th STOC, pp. 365-377, 1982.
4. Barany, I., Furedi, Z., "Mental Poker with Three or More Players," Technical Report, Mathematical Institute of the Hungarian Academy of Sciences (1983).
5. Heisenberg, W., "Physics and Beyond", translated by Arnold Pomerans, Harper & Row, 1971, p. 139.
6. Yung, M. M., "Cryptoprotocols: Subscription to a Public Key, The Secret Blocking and the Multi-Player Mental Poker Game", manuscript, 1984.
7. Fich, F., Goldwasser, S., "Sending Messages without Hidden Information and Applications to Multiperson Games," manuscript, 1984.
8. Evans, A., Kantrowitz, W., Weiss, E., "A User Authentication Scheme Not Requiring Secrecy in the Computer." CACM (August 1974) Vol. 17(8), pp. 437-442.
9. Rabin, M., "Digitalized Signatures," in Foundations of Secure Computation, DeMillo, Dobkin and Lipton, eds., Academic Press, 1978, pp. 155-168.
10. Chaum, D., "Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms," CACM (February 1981) Vol. 24(2), pp. 84-88.
11. Merritt, M., "Cryptographic Protocols," Ph.D. thesis, Georgia Institute of Technology (1983) GIT-ICS-83/06.