2005

Efficient TTP-free mental poker protocols

Weiliang Zhao
University of Western Sydney, wzhao@uow.edu.au

Vijay Varadharajan
Macquarie University

Publication Details

Zhao, W. & Varadharajan, V. (2005). Efficient TTP-free mental poker protocols. International Symposium on Information Technology: Coding and Computing (pp. 745-750). Australia: IEEE.

Efficient TTP-free Mental Poker Protocols

Weiliang Zhao
School of Computing and Information Technology
University of Western Sydney
NSW 2747, Australia
wzhao@cit.uws.edu.au

Vijay Varadharajan
Department of Computing
Macquarie University
NSW 2109, Australia
vijay@ics.mq.edu.au

Abstract

Zhao et al proposed an efficient mental poker protocol which did not require using a Trusted Third Party(TTP). The protocol is efficient and suitable for any number of players but it introduces a security flaw. In this paper, we propose two mental poker protocols based on Zhao's previous work. The security flaw has been removed and the additional computing cost is small.

1 Introduction

With the growth and popularity of the Internet, online gambling is becoming increasingly significant [10, 16]. Mental poker is one of the most popular games of online gambling and the fairness of the involved players is challenging from the view point of data security. Mental poker was firstly proposed by Shamir et al [13] in 1979 and many attempts have been made to achieve protocols that would allow people to play mental poker [8, 14, 9, 1, 15, 4, 5, 6].

Mental poker proposals can be categorized into two groups depending on whether a TTP is used or not. Mental poker protocols with a TTP [10, 8, 1] are normally simple and efficient. However, the assumption of a fully trusted TTP is not tolerable in online gambling. Mental poker protocols without TTP have been proposed [4, 5, 6, 11]. These protocols use zero-knowledge proof and the protocols are not efficient in the shuffling and dealing of cards. They have sound security but not practical in real implementation. Shamir et al [14] utilized commutative cryptosystems to develop their mental poker protocol without TTP but the protocol is limited to two players only.

In order to develop an efficient and secure mental poker protocol which can satisfy all the major requirements of a real poker protocol and can be used for the purpose of online gambling, Zhao et al [17] have proposed an efficient TTP-free mental poker protocol based on multiple encryption and decryption of individual cards. The protocol proposed by Zhao et al introduces a security flaw which was pointed out by Castellà-Roca et al [2]. In this paper, we propose two mental poker protocols based on the previous work of Zhao et al. The security flaw has been removed and the additional computing cost is small. The proposed mental poker protocols are TTP-free and efficient. The confidentiality of cards is achieved and the protocol is suitable for any number of players. The effect of collusion is minimum. If a Dealer is introduced, the strategies of players are confidential. The trust on the Dealer is limited to the confidentiality of the strategies only.

Section 2 describes a multiple encryption and decryption system which is the cornerstone of our TTP-free mental poker protocols. Section 3 describes the details of our mental poker protocols. The initialization of card games, shuffling of a set of cards and the dealing of cards in games are described. Section 4 discusses the related works. Section 5 gives an overview of the security properties of our protocols. Section 6 provides the concluding remarks.

2 Multi-Party Encryption and Decryption

In this section, we will discuss a multi-party encryption and decryption system based on the ElGamal cryptosystem. Without losing generality, we assume that there are two parties A and B. The two parties employ a common prime number p and have the key pairs:

$$\mathcal{K}_A = \{(p, \alpha_A, k_A, \beta_A) : \beta_A \equiv {\alpha_A}^{k_A} (\bmod\ p)\}$$
$$\mathcal{K}_B = \{(p, \alpha_B, k_B, \beta_B) : \beta_B \equiv {\alpha_B}^{k_B} (\bmod\ p)\}$$

In $\mathcal{K}_A$, k_A is the secret key and $\{p, \alpha_A, \beta_A\}$ is the public key. In $\mathcal{K}_B$, k_B is the secret key and $\{p, \alpha_B, \beta_B\}$ is the public key. The multiple encryption and decryption employ ElGamal's asymmetric cryptosystem [7]. The multiple encryption and decryption of message x are as follows:

- A chooses random number r_A, and the result of encryption of x with A's public key $\{p, \alpha_A, \beta_A\}$ has two parts y_{1A} and y_{2A}:

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)
0-7695-2315-3/05 $ 20.00 **IEEE**

$$y_{1A} = \alpha_A^{r_A} \bmod p$$
$$y_{2A} = x\beta_A^{r_A} \bmod p$$

- B chooses random number r_B and encrypts the ciphertext of A's encryption (actually B encrypts y_{2A}) with B's public key $\{p, \alpha_B, \beta_B\}$ and obtains the following two parts,

$$y_{1B} = \alpha_B^{r_B} \bmod p$$
$$y_{2AB} = x\beta_A^{r_A}\beta_B^{r_B} \bmod p$$

Actually, there is no difference whether A or B encrypts first; we will get the same ciphertext y_{1A}, y_{1B}, y_{2AB}.

- A uses his private key k_A to decrypt $\{y_{1A},\ y_{2AB}\}$:

$$d_{\mathcal{K}_A}(y_{1A},\ y_{2AB}) = \ y_{2AB}\ ({y_{1A}}^{k_A})^{-1} = \ y_{2B} \bmod p$$

- B uses his private key k_B to decrypt $\{y_{1B},\ y_{2B}\}$ and obtains x:

$$d_{\mathcal{K}_B}\ y_{2B}) = \ y_{2B}\ ({y_{1B}}^{k_B})^{-1} = \ x \bmod p$$

x is the original message.

Actually, there is no difference whether A or B decrypts first; we could use the following formula to express the whole multi-party decryption

$$d_{\mathcal{K}_A,\mathcal{K}_B}(\ y_{1A},\ y_{1B},\ y_{2AB}) =$$
$$y_{2AB}\ ({y_{1A}}^{k_A})^{-1}\ ({y_{1B}}^{k_B})^{-1} = x \bmod p$$

The most important characteristic for the above system is the commutativity of the multiple encryptions and decryptions. The order of the encryptions and decryptions will not change the result. The mental poker protocol proposed in next section will employ the above commutative cryptosystem.

3 Mental Poker Protocols

Our target is to design a mental poker protocol for multiple players to play fair on-line mental poker games. The mental poker protocol must provide fairness for the involved parties. The fair mental poker protocol should cover both the shuffling and dealing of the cards in a fair manner. All the involved players must be sure that nobody has stacked the deck in the shuffling and there is no unexpected information leak in the dealing. In our proposed protocol, there is not a trusted third party involved during the game. The proposed protocol in this paper focuses on the processes of shuffling and dealing the cards only and is suitable for any set of cards. Our protocol deals with cards one by one which is different from the protocols based on permutations of cards [8]. Without losing generality, we assume that there are two players Alice and Bob in the card game and there is no real difference when more players are involved.

3.1 Initialization

1. Alice and Bob agree to choose the same 52 tokens for 52 cards, that are suitable encoding set $\{1, ... , 52\}$.

2. Alice and Bob agree to choose the same prime number p.

3. Alice chooses her encryption and decryption key pairs as follows:

$$\mathcal{K}_A = \{(p, \alpha_A, k_A, \beta_A) : \beta_A \equiv {\alpha_A}^{k_A}(\bmod p)\}$$

4. Alice has a public/private key pair pka and ska, ska for the signature by Alice and pka for the verification of the Alice's signature by others.

5. Bob chooses his encryption and decryption key pairs as follows:

$$\mathcal{K}_B = \{(p, \alpha_B, k_B, \beta_B) : \beta_B \equiv {\alpha_B}^{k_B}(\bmod p)\}$$

6. Bob has public/private key pair pkb and skb, skb for the signature by Bob and pkb for the verification of the Bob's signature by others.

3.2 Protocol Description

In this section, we propose two mental poker protocols based on the cryptosystem with multiple encryptions and descriptions described in section 2. The first protocol referred as protocol A requires brand new encryption and decryption keys for every game. The second protocol referred as protocol B does not require brand new encryption and decryption keys for every game. In protocol A, the decryption key is published at the end of the game. In protocol B, the decryption key is secret at any time. The encryption and decryption keys can be reused. In both protocols, the card shuffling and dealing are based on the encryption/decryption of individual cards.

3.2.1 Protocol A

I. Card Shuffling

1. Alice chooses a set of secret random numbers $\{r_{A1}$, r_{A2}, ..., r_{A52} $\}$ and then encrypts original card n with encryption key $\{p, \alpha_A, \beta_A\}$ and random number r_{An} for each card in the card set $\{1, 2, ..., 52\}$. The set of encrypted cards is $\{E_A(1), ..., E_A(52)\}$ and the cards are put in the set with a random permutation. The set of cards is sent to Bob.

2. Bob chooses a set of secret random numbers $\{r_{B1}$, r_{B2}, ..., r_{B52} $\}$ and then encrypts card m of the encrypted card set by Alice with encryption key $\{p, \alpha_B, \beta_B\}$ and random number r_{Bm} for each card in the set of cards encrypted by Alice. The set of double encrypted cards is $\{E_{AB}(1), ..., E_{AB}(52)\}$ and they are put in the set with a random permutation. Bob signs the double encrypted cards one by one and sends them to Alice.

3. Alice signs the double encrypted cards one by one. The set of cards is $\{< E_{AB}(1) >_{ska,skb}, ..., < E_{AB}(52) >_{ska,skb}\}$. Alice sends them to Bob.

Now the deck of cards has been prepared. All the cards are encrypted by Alice and Bob with their signatures.

II. Card Dealing

There are 52 cards encrypted by both Alice and Bob. At the very beginning, the set of available order numbers is $\{1, ..., 52\}$. During the game, if some cards are in players' hands, the corresponding order numbers are deleted from the available set. When a player needs a card, the following protocol is carried out.

1. Alice needs to draw a card n, n is the card order after the double encryptions. She sends n and $< H(n) >_{ska}$ to Bob.

2. Bob checks Alice's signature and then checks that n is in the available set or not. If it is not in the available set, Bob sends Alice a suitable message. If it is in the available set, Bob decrypts the double encrypted card n. After Bob's decryption, it becomes $E_A(n)$. Bob sends $E_A(n)$, $< m, H(E_A(n)) >_{skb}$ to Alice. Bob deletes n from his available set.

3. Alice checks Bob's signature and decrypts $E_A(n)$ to open the card and adds the card to her hand. Alice deletes n from her available set.

III. Fairness Verification

At the end of the game, all the involved players publish their encryption/decryption keys and the players can verify that all players have played fairly.

3.2.2 Protocol B

I. Card Shuffling

1. Alice chooses a set of secret random numbers $\{r_{A1}$, r_{A2}, ..., r_{A52} $\}$ and then encrypts original card n with encryption key $\{p, \alpha_A, \beta_A\}$ and random number r_{An} for each card in the card set $\{1, 2, ..., 52\}$. The set of encrypted cards is $\{E_A(1), ..., E_A(52)\}$ and they are put in the set with a random permutation PA. Alice signs the set of hash of r_{An} $\{n = 1, 2, ..., 52\}$ to get $\{< H(r_{A1}) >_{ska}, < H(r_{A2}) >_{ska}, ..., < H(r_{A52}) >_{ska}\}$. Alice signs the hash of PA. Alice sends $\{E_A(1), ..., E_A(52)\}$, $< H(PA) >_{ska}$ and $\{< H(r_{A1}) >_{ska}, < H(r_{A2}) >_{ska}, ..., < H(r_{A52}) >_{ska}$ $\}$ to Bob.

2. Bob chooses a set of secret random numbers $\{r_{B1}$, r_{B2}, ..., r_{B52} $\}$ and then encrypts card m of the encrypted card set by Alice with encryption key $\{p, \alpha_B, \beta_B\}$ and random number r_{Bm} for each card in the set of cards encrypted by Alice. The set of double encrypted cards is $\{E_{AB}(1), ..., E_{AB}(52)\}$ and they are put in the set with a random permutation PB. Bob signs the set of hash of r_{Bm} $\{m = 1, 2, ..., 52\}$ to get $\{< H(r_{B1}) >_{ska}, < H(r_{B2}) >_{ska}, ..., < H(r_{B52}) >_{ska}\}$. Bob signs the double encrypted cards one by one and signs the hash of PB. Bob sends $\{< E_{AB}(1) >_{skb}, ..., < E_{AB}(52) >_{skb}\}$, $< H(PB) >_{skb}$ and $\{< H(r_{B1}) >_{skb}, < H(r_{B2}) >_{skb}$ $, ..., < H(r_{B52}) >_{skb}\}$ to Alice.

3. Alice put her signature on each card in $\{< E_{AB}(1) >_{skb}, ..., < E_{AB}(52) >_{skb}\}$. The set of cards becomes $\{< E_{AB}(1) >_{ska,skb}, ..., < E_{AB}(52) >_{ska,skb}\}$. Alice sends the doubled signed cards to Bob.

Now the deck of cards has been prepared. All the cards are encrypted by Alice and Bob with their signatures.

II. Card Dealing

The card dealing of protocol B is exactly the same as the card dealing in protocol A. All details of card dealing have been provided in protocol A and we will not repeat it again here.

III. Fairness Verification

At the end of the game, all the involved players publish their random permutations and set of random numbers that they have used in the encryptions of cards in the card shuffling. The players can use the encryption keys and the random numbers used in the encryptions to check that all players have played fairly or not.

4 Related Works

The protocols proposed in this paper are closely related to Shamir et al mental poker [14] and Zhao et al mental poker [17]. Both of them will be described in this section.

4.1 Shamir et al mental poker

Shamir et al [14] proposed a mental poker protocol based on RSA cryptosystem. Alice and Bob are the two involved parties. E_A and D_A are Alice's encryption and decryption functions; E_B and D_B are Bob's encryption and decryption functions respectively. For a message x, $E_A(D_B(x)) = D_B(E_A(x)); E_B(D_A(x)) = D_A(E_B(x)); E_A(E_B(x)) = E_B(E_A(x)); D_A(D_B(x)) = D_B(D_A(x))$. The above relations show the commutativity of the encryptions and decryptions in the cryptosystem. Another important characteristic of the cryptosystem is that the encryption key and decryption key are both secret in the game. The mental protocol is as follows:

1. Alice encrypts each card in a deck of cards $\{1, ..., 52\}$ separately and permutes the set in a random order. Alice sends the set $\{E_A(1), ..., E_A(52)\}$ to Bob.
2. Bob chooses five encrypted cards at random, for example $\{E_A(6), E_A(8), E_A(17), E_A(25), E_A(33)\}$, and sends them to Alice, Alice can decrypt them and know that they are $\{6, 8, 17, 25, 33\}$.
3. Bob chooses five different encrypted cards, for example $\{E_A(3), E_A(11), E_A(19), E_A(23), E_A(41)\}$, encrypts them with his secret key and sends them back to Alice in a randomly ordered set $\{E_B(E_A(3)), E_B(E_A(11)), E_B(E_A(19)), E_B(E_A(23)), E_B(E_A(41))\}$.
4. Alice decrypts cards one by one and sends Bob the resulting set $\{E_B(3), E_B(11), E_B(19), E_B(23), E_B(41)\}$. Bob can decrypt and know that they are $\{3, 11, 19, 23, 41\}$.
5. At the end of the game, they could exchange their encryption keys and verify that all players have played fairly.

Lipton [12] analyzed the above proposal and found that there was at least one bit of information leak. The information leak comes from the judgement of quadratic residue on the number which stands for a card in the mental poker protocol. For a number x, if $x \equiv y^2 (mod\ n)$ for some y, x is a quadratic residue modulo n; otherwise, x is non-quadratic residue. All keys must be odd numbers, and $x^k (mod\ n)$ is a quadratic residue if and only if x is. If the players know a card is quadratic residue or not, they can know one bit information about the card based on that the encrypted card is quadratic residue or not. There is no guarantee to get rid of the one bit information leak even the above mental poker protocol modified based on Lipton's suggestions [3].

4.2 Zhao et al mental poker

Zhao et al [17] proposed a mental poker proposal based on the ElGamal's cryptosystem with the commutativity of multiple encryptions and descriptions. In the protocol, Alice and Bob are the players and they have key pairs:

$$\mathcal{K}_A = \{(p, \alpha_A, k_A, \beta_A) : \beta_A \equiv {\alpha_A}^{k_A} (\bmod\ p)\}$$
$$\mathcal{K}_B = \{(p, \alpha_B, k_B, \beta_B) : \beta_B \equiv {\alpha_B}^{k_B} (\bmod\ p)\}$$

Alice and Bob agree with that the card deck is represented by 52 tokens $\{x_1, x_2, ..., x_{52}\}$. The protocol is as follows:

1. Alice chooses a secret random number r_A, and then encrypts original cards one by one. The set of encrypted cards is $\{E_A(1), ..., E_A(52)\}$ in a random order. Alice signs the hash function of r_A to get $< H(r_A) >_{ska}$. Alice sends $\{E_A(1), ..., E_A(52)\}$ and $< H(r_A) >_{ska}$ to Bob.
2. Bob chooses a secret random number r_B, and then encrypts original cards one by one. The set of encrypted cards is $\{E_B(1), ..., E_B(52)\}$ in a random order. Bob signs the hash function of r_B to get $< H(r_B) >_{skb}$. Bob sends $\{E_B(1), ..., E_B(52)\}$ and $< H(r_B) >_{skb}$ to Alice.
3. Alice encrypts the set of cards encrypted by Bob and gets $\{E_{AB}(1), ..., E_{AB}(52)\}$. Alice sends the results to Bob.
4. Bob encrypts the set of cards encrypted by Alice and gets $\{E_{BA}(1), ..., E_{BA}(52)\}$. Bob sends the results to Alice.
5. Alice checks two sets of double encrypted cards with a different encryption order. If the two sets are not equal, then the protocol will be stopped. If they are equal, Alice signs the double encrypted cards one by one. With the notation $C[n] = E_{AB}(n)$ where $(n = \{1, ..., 52\}$ is the order number of cards, Alice gets $\{< H(C[1]) >_{ska}, ..., < H(C[52]) >_{ska}\}$. Alice signs the order of cards and gets $< C[1], ..., C[52] >_{ska}$. Alice sends the double encrypted cards, signatures of cards and signed order of cards to Bob.

6. Bob checks the set of double encrypted cards and their signatures by Alice. Bob checks two sets of double encrypted cards with a different encryption order. If the checks are successful, Bob signs double encrypted cards again and gets $\{< H(C[1]) >_{ska,skb}, ..., < H(C[52]) >_{ska,skb}\}$. Bob signs the order of cards again and gets $< C[1], ..., C[52] >_{ska,skb}$. Bob sends signatures of cards and signed order of cards to Alice.

Now the deck of cards has been prepared. All the cards are encrypted by Alice and Bob with their signatures. At the very beginning, the set of available cards is $\{1, ..., 52\}$. During the game, if some cards are in players' hands, the corresponding order numbers are deleted from the available set. When a player needs a card, the following protocol is carried out.

1. Alice needs to draw a card m, m is the card order after the double encryptions. She sends m and $< H(m) >_{ska}$ to Bob.

2. Bob checks Alice's signature and then checks that m is in the available set or not. If it is not in the available set, Bob sends Alice a suitable message. If it is in the available set, Bob decrypts the double encrypted card m. The original order of the card is n, the card m is $C[n]$. After Bob's decryption, it becomes $E_A(n)$. Bob sends $E_A(n)$, $< m, H(E_A(n)) >_{skb}$ to Alice. Bob deletes m from his available set.

3. Alice checks Bob's signature and decrypts $E_A(n)$ to open the card and adds the card to her hand. Alice deletes m from her available set.

When the game is over, Alice and Bob reveal their secret random number r_A and r_B. Both Alice and Bob can check whether the other party has been cheating or not.

The above protocol is computationally efficient and fast. Unfortunately, the protocol introduces a security flaw. The nature of ElGamal encryption is to multiply the original message by a hiding factor. In the above protocol, a player uses the same random number to encrypt all cards. All cards are encrypted by multiplying the same hiding factor. The security flaw of the protocol is that the common hiding factor could be calculated with the method introduced by Castellà-Roca et al [2].

5 Discussion

In this section, we discuss some important properties of mental poker protocols proposed in this paper. We compare our protocols with previously published protocols. The discussions are related to the details of proposed protocols in section 3, Shamir et al mental poker and Zhao et al mental poker in section 4.

5.1 Confidentiality of Cards

In order to design mental poker based on encryptions and decryptions of individual cards, Shamir et al designed a mental poker protocol based on RSA cryptosystem [13, 14]. Shamir et al mental poker is efficient but there is at least one bit information leak [12, 3]. To avoid the information leak, Zhao et al [17] proposed an efficient TTP-free mental poker based on ElGamal cryptosystem. Unfortunately, Zhao's mental poker introduced a security flaw [2]. The security flaw comes from the reusing of the same random number in the encryptions of the whole set of cards. In the proposed protocols in this paper, different random numbers are used in the encryptions of the set of cards. The mentioned security flaw is removed and the other characteristics of Zhao's mental poker have been kept. The confidentiality of cards is achieved.

5.2 Third Parties in Mental Poker

There are different trust assumptions for the involved third party. If the third party knows the card information in the shuffling and dealing of cards, we call it Card Salesman. If the third party does not know the card information in the shuffling and dealing of cards, we call it Dealer. The Dealer can know the card information after the game. There are some mental poker protocols with a Card Salesman [10, 8, 1]. The Card Salesman is a trusted third party and he will cause serious trust and security issues when mental poker is used in real gambling. There is a strong desire to get rid of Card Salesman and design real TTP-free mental poker. The protocols proposed in this paper have achieved the above requirements. A Dealer may be involved in mental poker for the confidentiality of strategy of players. If a mental poker requires players to reveal all information to peers at the end of game, the strategies of players are published and it is impossible for the players to bluff. If a Dealer is involved, the above situation can be changed. The Dealer will check the fairness of the running of the mental poker at the end of a card game. The Dealer does not know the cards in the shuffling and dealing but he is able to check the fairness of the whole game at the end. The Dealer is the only person who can know the strategy of each player. In the process of shuffling and dealing of protocol A and protocol B, all the information for checking will be sent to the Dealer instead of peer players. At the end of the game in protocol A, the Dealer is the only one who receives the encryption/decryption keys and verify that all players have played fairly. At the end of the game in protocol B, all the involved players reveal their random permutations and the set of random numbers to the Dealer only. The Dealer can use the encryption keys and the random numbers to check that all players have played fairly or not.

5.3 Multiple Players and Collusion

The protocols proposed in this paper are based on a cryptosystem with the commutativity of multiple encryptions and decryptions. It is convenient to expand these protocols to multiple players. In a card game, cards are encrypted by all the players and the protocols can achieve minimal effect of collusion. When some players collude, they can not get more information than the cards on their hands. A card can be opened only when all players have decrypted it. Any subset of players can not know anything about the cards of other players. No collusion can get information about cards untouched and cards in the hands of honest players.

5.4 Clarity and Efficiency

The protocols proposed in this paper are based on multiple ElGamal encryptions/decryptions. The protocols are simple and clear from the view point of understanding. The protocols are efficient as well. If there are n players in a game, there are maximum $52 \times n$ ElGamal encryptions and $52 \times n$ ElGamal decryptions. In protocol A, a new game needs a new encryption/decryption key pair. In protocol B, the encryption/decryption key pair can be reused in multiple games. The fairness checking in protocol A is easier than that in protocol B. To compare with Zhao's mental poker [17], the additional computing cost is that each player needs to generate 52 random numbers instead of one random number. The increasing computing cost is quite limited.

6 Conclusion

The mental poker protocols proposed in this paper are updated versions of Zhao's mental poker [17]. The security flaw [2] in Zhao's mental poker has been removed by using different random numbers for the encryption of each card in card shuffling. The proposed protocols are secure, efficient and are suitable for any number of players. To compare with Zhao's mental poker, the additional computing cost of proposed protocols is quite small. The protocols have got rid of the Card Salesman who knows all the cards when the card game is being played and the collusion of players is limited to revealing the cards on the hands of cheating players. There is no third party involved in protocol A and protocol B. If a Dealer is introduced in these protocols, the strategies of players become confidential to peer players. The Dealer does not know the cards in the shuffling and dealing but he knows the strategies of players after the game.

The Internet has become an important marketplace for online gambling. Card games are popular for people to gamble over the Internet. The protocols proposed in this paper are suitable for the purpose of online gambling [10, 16].

References

[1] I. Barany and Z. Furedi. Mental poker with three or more players. Technical report, Technical Report, Mathematical Institute of the Hungarian Academy of Science, 1983.

[2] J. Castella-Roca and J. Domingo-Ferrer. On the security of an efficient ttp-free mental poker protocol. In *Information Technology: Coding and Computing*, volume 2, pages 781–784, 2004.

[3] D. Coppersmith. Cheating at mental poker. In *Proceedings of Advances in Cryptology - CRYPTO'85*, pages 104–107. Springer-Verlag, 1985.

[4] C. Crepeau. A secure poker protocol that minimizes the effect of player coalitions, 1986.

[5] C. Crepeau. A zero-knowledge poker protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face. In *Proceedings of Advances in Cryptology - CRYPTO'86*, pages 239–247. Springer-Verlag, 1986.

[6] C. Crepeau and J. Killian. Discrete solitary games. In *Proceedings of Advances in Cryptology - CRYPTO'93*, pages 319–330. Springer-Verlag, 1994.

[7] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transaction on Information Theory*, 31:469–472, 1985.

[8] S. Fortune and M. Merrit. Poker protocols. In *Proceedings of Advances in Crytology - CRYPTO'84*, pages 454–464. Springer-Verlag, 1985.

[9] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14th ACM Symposium on the Theory of Computing*, pages 270–299, 1982.

[10] C. Hall and B. Schneier. Remote electronic gambling. In *Proceedings of 13th Annual Computer Security Applications Conference*, pages 232–238, 1997.

[11] K. Kurosawa, Y. Katayama, and W. Ogata. Reshuffable and laziness tolerant mental card game protocol. *TIEICE:IEICE Transactions on Communications/Information and Systems*, E00-A, 1997.

[12] R. Lipton. How to cheat at mental poker. In *Proceedings of the AMS Short Course in Cryptography*, 1981.

[13] A. Shamir, R. Rivest, and L. Adleman. Mental poker. 1979. MIT/LCS/TM-125, Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

[14] A. Shamir, R. Rivest, and L. Adleman. Mental poker. *Mathematical Gardner,*, pages 37–43, 1981.

[15] M. Yung. Cryptoprotocols: Subscriptions to a public key, the secret blocking, and the multu-player mental poker game. In *Proceedings of Advances in Cryptology - CRYPTO'84*, pages 439–453. Springer-Verlag, 1985.

[16] W. Zhao, V. Varadharajan, and Y. Mu. Fair on-line gambling. In *Proceedings of 16th Annual Conference of Computer Security Applications*, pages 394–400, 2000.

[17] W. Zhao, V. Varadharajan, and Y. Mu. A secure mental poker protocol over the internet. In *Proceedings of Australian Information Security Workshop*, volume 21, pages 105–109, Adelaide, Australia, 2003. Australian Computer Society.