2006

Fair online gambling scheme and TTP-free mental poker protocols

Weiliang Zhao
University of Wollongong, wzhao@uow.edu.au

Vijay Varadharajan
Macquarie University

Publication Details

Zhao, W. & Varadharajan, V. (2006). Fair Online Gambling Scheme and TTP-free Mental Poker Protocols. Journal of Information Assurance and Security, 1 (2), 95-106.

Fair Online Gambling Scheme and TTP-free Mental Poker Protocols

Weiliang Zhao[1] and Vijay Varadharajan[2]

[1]School of Computing and Information technology
University of Western Sydney, NSW 1797, Australia
wzhao@cit.uws.edu.au

[2]Department of Computing
Macquarie University, NSW 2109, Australia
vijay@ics.mq.edu.au

Abstract: Online gambling allows users to participate in mental games and bet over the Internet. It can open new opportunities for casinos to make more money; however, at the same time, there are a lot of security challenges. In this paper, we describe our scheme for secure online gambling and secure mental poker protocols over the Internet based on our previous researches. Our scheme provides both the fairness of online gambling processes and secure linking of the online gambling with payment. A unique link between payments and gambling outcome is provided so that the winner can be ensured to get the payment. The gambling games are generic in our fair online gambling scheme. Mental poker is one of most popular games of online gambling. In this paper, efficient and secure mental poker protocols are provided. These mental poker protocols are based on multiple encryption and decryption of individual cards. These protocols satisfy all major security requirements of a real mental poker. The card salesman has been got rid of and the minimal effect due to collusion has been guaranteed. These protocols are more efficient compared with other known protocols. The strategies of players can be kept confidential with the introduction of a dealer. These mental poker protocols are suitable for implementation in an online card game.

Keywords: Online Gambling, Mental Poker Protocols, Fair Exchange, Applied Cryptography.

1 Introduction

The Internet has become an important marketplace for online gambling. There are numerous online gambling and casino web sites on the Internet. Growth in Internet gambling is expected all around the globe, as the universality of the Internet combines with a worldwide interest in gambling. The Internet paradigm could provide a significant cross-marketing opportunity for traditional operators, particularly to the young generation who are more familiar with the Internet. The term online casino is broadly used to refer to web sites that can offer its services as a casino over open networks, such as the Internet. On many sites, customers can play against the site operators (dealers) or against other customers. When we consider online gambling, the bets and payoffs are the two most important issues to be addressed. With the development of cryptographic research, many practical solutions have been proposed about gambling over the Internet [1][2][3] and how to pay money over the Internet [4][5][6][7][8][9].

The research about different kinds of online games has a long history, particularly regarding how to play poker in an online environment. A scheme for playing "Mental Poker" was proposed by Shamir, Rivest and Adleman [10] in 1979. Following this, many attempts have been made to achieve protocols that would allow people to play "Mental Poker". With the growth and popularity of the Internet, on-line gambling is becoming increasingly significant [1][11]. Mental poker is one of most popular games of online gambling. For the purpose of on-line gambling over the Internet, additional requirements on a poker protocol need to be considered. The need for secure and efficient protocols for card games is becoming more and more necessary.

There have been several protocols based on public-key cryptography described in literature for playing poker [12][13][14][15][16][17][18]. These protocols require that players generate new key pairs for each game they play, and this could be computationally intensive. Many of these protocols are not secure in their implementations, and they leak partial information about the cards themselves. There are some protocols based on multiple permutations which require a trusted card salesman to be involved in the games [1][15][19]. If card games are used for the purpose of online gambling, the assumption of a fully trusted card salesman is not tolerable. Some protocols [20][21][22] have no information leakage and meet many of the important requirements of a real poker game, but they are not practical in their implementations. They use zero-knowledge proofs and are not efficient in shuffling and dealing with cards.

We are interested in the situations where the online casino is not necessarily trusted [23]. That is, we have "untrusted" gaming sites. This is particularly important in practice as in many countries online gambling is not regulated by government authorities. In such cases, for instance, there may not be any guarantee that the casino authorities are not having an unfair advantage over the players. In such circumstances, at least as far as the playing of the game is concerned, it is necessary to have fair exchange schemes. A fair exchange scheme [24][25][26][27] requires a trusted third party (TTP) who helps to resolve disputes amongst the playing entities. In general, TTP can be online or offline. For efficiency reasons, it is preferable that TTP is offline. In this case, TTP only comes into play when a problem occurs in the gambling system; otherwise, TTP is not contacted. Based on our previous research [11], we describe in this paper a general fair exchange protocol with credit card payment and a fair online gambling scheme. The contributions of our fair online gambling scheme are twofold. Firstly, it involves the proposal of a fair exchange scheme for online gambling processes. Secondly, it provides a secure linking of the online gambling with payment. Hence the overall scheme we propose provides a unique link between a gambling process and its associated payment, which makes the whole gambling process fair.

For the games themselves, we are interested in efficient and secure online card games which can satisfy all major requirements of a real poker protocol. Based on our previous research [28][29], we provide mental poker protocols based on multiple encryptions and decryptions of individual cards. These protocols have complete

confidentiality of cards and are efficient in real implementations. They are suitable for any number of players to play card games for the purpose of on-line gambling. The effect of collusion is the minimum and the strategies of players are confidential with the introduction of a dealer.

Section 2 is about the fair online gambling scheme. Subsection 2.1 discusses the security requirements and assumptions of fair online gambling. Subsection 2.2 provides the details of the protocol of general fair exchange with credit card payment. Subsection 2.3 describes the online gambling scheme for games from authorized organizations. Subsection 2.4 describes the online gambling scheme for pure luck games. Section 3 is about the efficient mental poker protocol. Subsection 3.1 reviews the most popular protocols of mental poker. Subsection 3.2 describes the cryptosystem with multi-party encryption and decryption and discusses the commutativity of multiple encryption and decryption. Subsection 3.3 provides the details of our mental poker protocols, which contain the system initialization, card shuffling and card dealing. Subsection 3.4 discusses several important security issues of mental poker protocols and compares our mental poker protocols with well-known protocols. Finally, section 4 concludes the paper with some final remarks. **Appendix A** provides the details of equality proof of knowledge. **Appendix B** provides the details of proof of equivalence of discrete logarithm to discrete log-logarithm(PEDLDLL).

2 Fair Online Gambling Scheme

2.1 Security Requirements and Assumptions

For online gambling, physical cheating in traditional gambling environments can be easily removed. For example, players in card gambling games can not hide cards, mark cards etc and players can not collaborate with dealers. However, in online gambling, there are many new kinds of security problems. Here, we list some of the most important ones:

- hackers in open network environment
- cheating by players
- cheating by casinos
- unauthorized use of credit cards
- collusion among players and casinos
- fairness of the gaming processes
- fairness of the payments
- linkage between payments and gaming results
- privacy information collections (gambling habits and strategies)

Actually, the above security issues are related to each other, and a successful scheme as overall solution for online gambling will need to address all of them. However, an individual scheme may emphasize on some aspects of these requirements based on different kinds of gambling processes. Currently, many of existing online casino games provide some level of security and privacy. However, most of them are entirely based on the trust of the casinos. The players have no way to check the fairness of games. Also, the fairness of payments may not be guaranteed. The players and casinos are adversary parties in the gambling. Hence it is undesirable to trust the casinos by default. We are interested in the situations where the online casinos are not trusted. This is particularly important in practice as in many countries that online gambling is not regulated by government authorities. In such cases, there may not be any guaranty that the casino authorities are not having an unfair advantage over the players. In such circumstances, at least as far as the playing of the game is concerned, it is necessary to have suitable schemes with fairness. This kind of schemes must address both the fairness of gambling processes and payments. Gambling processes and their associated payments must be uniquely linked with each other.

The natures of processes and security requirements vary for different kinds of games in online gambling. To guarantee the fairness of gaming processes, a fair online gambling scheme needs to consider the details of the games themselves. In online gambling, many games could come from some authorized organizations and they could be regarded as standard gambling games. The set up of these kinds of games should strictly follow the rules defined by the authorized organizations. The players and dealers must both agree with the rules of a game. Another popular kind of games are those of chance games which are referred to as pure luck games. The players will bet on the output of the games; actually they are betting on lucky numbers. In our fair online gambling scheme, we will only discuss these two kinds of generic games: one is the games from an authorized organization and the other one is pure luck games.

An online gambling scheme must be associated with an online payment scheme. Credit-based payment methods [8] are quite popular in online gambling casinos. For our online gambling scheme, we will only consider the credit card based payment. The fair exchange scheme we propose resolves the following disputes: (1) the dealer refuses to make a payment to the player who has won, (2) the dealer denies a payment that was made by a player in advance, and (3) the player, who paid to the dealer in advance, refuses to accept the gambling outcome after he or she has lost.

There are different kinds of online gambling in the real world, however there are some general characteristics for all gambling systems. We will only consider some generic processes of online gambling. In our scheme for online gambling, we make the following assumptions :

a) Two-way payments are involved.
   — Anonymous.
   — Credit Card Payment.
b) Bank is offline.
c) Trusted Third Party (TTP) is offline.
d) Cheating is prevented during whole process.
   — Information must be checked.
   — If there is a dispute, TTP will resolve it.

In this paper, we will discuss both games from authorised organisations and pure luck games respectively. General fair exchange and electronic payment are the bases of a fair online gambling scheme. In the following section, we will give the details of fair exchange protocol with credit card payment.

2.2 General Fair Exchange with Credit Card Payment

Based on the research progresses of generic fair exchange protocols [26][27, 30][31] and credit based payment for electronic commerce [8], we propose a fair exchange protocol with credit payment using the technique of Equality Proof Knowledge (Appendix A) and PEDLDLL (Proof of Equivalence of Discrete Logarithm to Discrete Log-logarithm, Appendix B). Both the TTP and the bank (the financial institution for credit authority) are offline. The credit information of the client is anonymous. The general fair exchange protocol with credit payment is the cornerstone of our fair online gambling scheme.

2.2.1 Notations

Here we give the general notations which will be used in the description of our general fair exchange protocol.

(1) Parties:

- C: Client
- M: Merchant
- TTP: Trusted Third Party
- B: Bank (Financial Institute for Credit Authority)

(2) Public Key Cryptosystems:

- PKX: Public key of user X.
- SKX: Private key of user X.
- $P_{enc}(PKX, m)$: Encryption of message m with public key PKX.
- $P_{dec}(SKX, c)$: Decryption of ciphertext c with private key SKX.

(3) Digital Signature Schemes:

- pkx: Verifying key of user X.
- skx: Signing key of user X.
- $< m >_{skx}$: Creation of signature of m under signing key skx.
- $S_{veri}(pkx, < m >_{skx}, m)$: Verification of signature $< m >_{skx}$ on message m, *true* for valid and *false* for invalid.

(4) Other items:

- t_x : Timestamp generated by party X.
- $H(m)$: Hash function on message m.

2.2.2 System Setup

There are four parties in our protocol, and they are Client, Merchant, TTP and Bank. Client has a pair of public and private keys: PKC and SKC, and a pair of signing and verifying keys: skc and pkc. Dealer has a pair of public and private keys: PKD and SKD and a pair of signing and verifying keys: skd and pkd. TTP has a pair of public and private keys: PKT and SKT. We will employ the technique of proof of equivalence of discrete logarithm to discrete log-logarithm. The above key pairs must follow some overall rules of the whole system. This means that these key pairs must be set up based on the same set of algorithms and parameters. If necessary, the signature scheme of TTP, public key cryptosystem of bank and signature scheme of bank can be defined independently. They need not follow the same set of algorithms and parameters.

At first, we choose three primes to set up the system. The three primes are p, q and q', which are of the form $p = 2q+1$ and $q = 2q'+1$. We will use ElGamal cryptosystem for encryption and decryption and a DSA-like scheme for signature.

Public Key Cryptosystems
q is the prime number for the ElGamal cryptosystem. Z_q^* is a intractable multiplicative group with order $q-1$. G is a generator of Z_q^*. SKX is the private key and PKX is the public key. $PKX = G^{SKX} \mod q$ and $SKX \in \{1, 2, \ldots, q-2\}$.

The ciphertext of m under PKX is:

$$cx = P_{enc}(PKX, m) = (W, V)$$

where $W = G^w \bmod q$ and $V = m(PKX)^w \bmod q$, w is randomly chosen from $\{1, 2, \ldots, q-2\}$.

The message after decryption is:

$$m = V \cdot W^{-SKX} \mod q$$

Digital Signature Scheme
p is the prime number for the DSA-like digital signature scheme. Z_p^* is a intractable multiplicative group with order $p-1$. g is a generator of Z_p^*. skx is the signing key and pkx is the verifying key. $pkx = g^{skx} \mod p$ and $skx \in \{1, 2, \ldots, q-2\}$.

The signature of m under pkx is :

$$< m >_{skx} = (r, s)$$

where $r = g^k \bmod p$ and $s = k^{-1}(h(m) + r \cdot skx) \bmod q$. k is randomly chosen from $\{1, 2, \ldots, q-2\}$ and $h(\ldots)$ is the hash function.

For verification of signature, $S_{veri}(pkx, < m >_{skx}, m)$ is to check

$$r^s \stackrel{?}{=} g^{h(m)} \cdot (pkx)^r \mod p$$

2.2.3 Construction of Important Tokens

In this section, we will give the details of digital tokens used in our fair exchange protocol with credit card based payment.

(1) Credit Card
The token for credit card is of the form

$$\mathcal{C} = < C, l, h_1, h_2, \cdots, h_l, E, A >_{skb}$$

The credit token contains the client's identity C, the confidence level l, the expiry date E, maximum credit amount A and $h_i = g_i^x \mod p$, where $g_i \in Z_p^*$ are common generators for $i = 1, 2, \cdots, l$, where x is the concatenation of PIN number, credit card number and salt. The credit token is signed by the bank using its private key skb.

(2) Payment Slip
The data in the payment slip is

$$SlipData = \mathcal{C}, M, O, \$, tc,\ H(\mathcal{C}, M, O, \$, tc),$$

where M is ID of merchant, O is the order, \$ is the amount of money and currency type and t_c is the timestamp generated by the client C.

The payment slip token has the form

$$Slip =< SlipData >_{skc},$$

The payment slip is signed by the client with private key skc.

(3) Encrypted Payment Slip
The encrypted payment slip token is

$$C_S = P_{enc}(PKT, Slip).$$

The client's payment slip is encrypted under the TTP's public key PKT. If necessary, TTP can open it with its private key SKT.

(4) Certificate of Encrypted Payment Slip
C_SCert is the token to prove C_S is a ciphertext of S without disclosing the signature. Here, we will give all the details of construction C_S and C_SCert. p and q are the two prime numbers used in our system. The client has a pair of signing key and verifying key $\{skc, pkc\}$, g is a generator of Z_p^* and $pkc = g^{skc} \mod p$. The TTP has public key and private key $\{PKT, SKT\}$, G is a generator of Z_q^* and $PKT = G^{SKT} \mod q$.

For encryption of message m, we have the following:

$$P_{enc}(PKT, m) = (W, V) \mod q,$$

where $W = G^w$ and $V = m(PKT)^w$, $w \in \{1, 2, \cdots, q-2\}$ is a randomly chosen number.

The signature scheme works as follows: Choose a random $k \in Z_q^*$, the signature has the form

$$Slip =< SlipData >_{skc} \equiv (r, s)$$

where $r = g^k \mod p$ and $s = k^{-1}(H(m) + r \times skc) \mod q$ and $pkc = g^{skc} \mod p$. $Slip$ is the payment slip.

Encrypting the above payment slip $Slip$ with PKT, we have, $P_{enc}(PKT, Slip) = (W, V)$. The encrypted payment slip with signature is then given as follows:

$$C_S = \{r, W, V\},$$

where $W = G^w \mod q$, $V = s(PKT)^w \mod q$.

With transformation $x = G$, $y = W^{-1} \mod q$, $z = PKT$, $X = r^V \mod p$, $Y = g^{H(S)}(pkc)^r \mod p$ and $\alpha = -w$, choose $w_i \in \{1, 2, \cdots, q-2\}$, then

$$t(x_i) = x^{w_i} \mod q,\ t(X_i) = X^{z^{w_i}} \mod p$$

and

$$c = H_l(x||y||z||X||Y||t(x_1)||t(X_1)||\cdots||t(x_l)||t(X_l))$$
$$c = c_1 c_2 \cdots c_l$$
$$r_i = w_i - c_i\alpha \mod q - 1$$

(R, c) is the certificate $C_S Cert$ for C_S.

The process of verification is to check,

$$c = H_l((x||y||z||X||Y||u_1||U_1||\cdots||u_l||U_l)$$

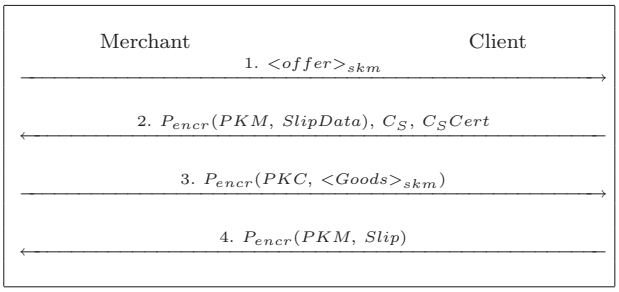
where $u_i = x^{r_i} y^{c_i} \mod q$, and

$$U_i = \begin{cases} X^{z^{r_i}} \mod p & if \ \ c_i = 0 \\ Y^{z^{r_i}} \mod p & if \ \ c_i = 1 \end{cases}$$

2.2.4 Fair Exchange Protocol

Based on the tokens defined in the last subsection, the fairness of the exchange between a client and a merchant can be achieved using the following fair-exchange protocol,

Fair Exchange Protocol



For the above protocol, if both the client and the merchant perform properly, the TTP will not be involved. The details of the protocol are as the followings:

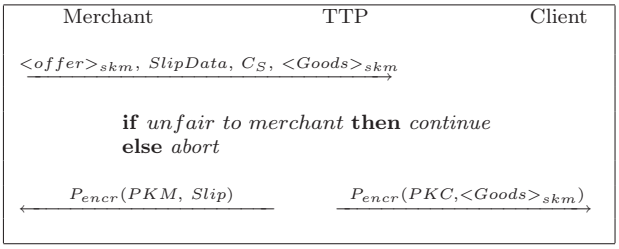
1. In step one, the merchant sends his signed offer to the client. The $offer$ should contain the description of the *Goods* and related trading information, such as price, valid date etc. The client checks the $offer$, and if the client is not satisfied with the $offer$, he can quit the protocol, and therefore it is fair for both parties.
2. In step two, the client sends the merchant his credit card C, order information O, amount of money and currency type \$ and time stamp t_c, encrypted payment slip C_S and the certificate $C_S Cert$. The encrypted payment slip C_S is encrypted with TTP's public key. The merchant checks the validity of the above data, and especially, the credit information and encrypted payment slip.

   (1) The merchant checks credit information with equality proof of knowledge (details are described in **Appendix A**.

   (2) The merchant uses $C_S Cert$ to check C_S is the ciphertext of the payment slip *Slip* signed by the client (details are described in section 2.2.3).

   If the merchant finds anything wrong in the above verification, he will quit the protocol, and the protocol is fair for both parties.
3. In step three, the merchant sends $P_{encr}(PKC, < Goods >_{skm})$ to the client. If the *Goods* is consistent with the $offer$, the client will continue the protocol. If the *Goods* is inconsistent with the $offer$, the client quit the protocol. If the merchant believes that it is not fair, he need to require TTP to run the resolve protocol.
4. In step four, the client sends $P_{encr}(PKM, Slip)$ to the merchant. If the merchant can not get the payment, the merchant will ask TTP to run resolve protocol.

The resolve protocol is normally initialized by the merchant as followings:

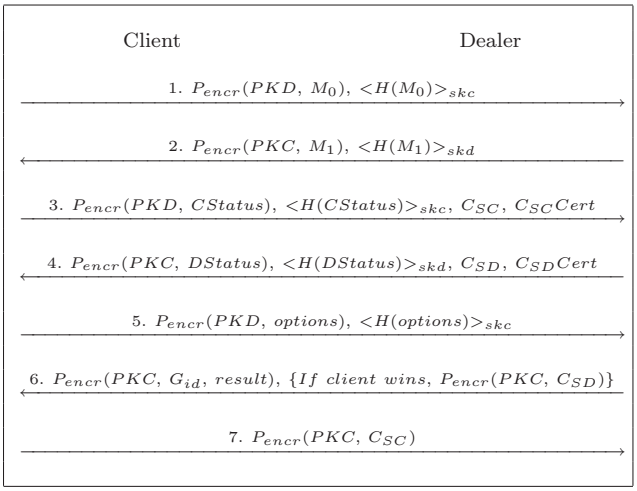


The above resolve protocol can guarantee the protocol to be fair with any case. For the client, if something is wrong, he can quit the protocol after step three and the whole protocol is fair. For the merchant, if something is wrong after step three, he can bring $offer$, *SlipData*, C_S, *Goods* to TTP. TTP will check the status. If it is really unfair to merchant, TTP will send the *Goods* to the client and send the *Slip* to the merchant.

2.3 Games From Authorized Organizations

In this section, we assume that games are from authorized organizations. The most important assumption with online casino is that casino itself should not be trusted. Both the client and the dealer must keep some secrets of their choice before they have given their bets. We have developed the protocol as follows:

Protocol of Gambling from Authority



1. The client chooses a game and sends the message to the dealer

   $$M_0 = (GName,\ C,\ tr_c)$$

   where $GName$ is the name of the game to play, C is the client ID, tr_c is *timestamp* for the message.
2. the dealer prepares the game for the client

   $$M_1 = (Game, P_{encr}(PKT, < Gnmae,\ G_{ID} >_{skd}), tg_c)$$

   Game is the executable program. G_{ID} is the default parameter for the game to run, tg_c is the timestamp for the preparation of the game. For the client, G_{ID} is the secret until the end of the game. $< Gnmae,\ G_{ID} >_{skd}$ is encrypted with TTP's public key PKT. If something is wrong, TTP can decrypt it and get the G_{ID}.
3. The client runs the game, gives his option and prepares his payment slip for betting.

   (a) The client gives his option

$$GStatus = C,\ M_0,\ M_1,\\ P_{encr}(PKT,\ <option>_{skc})$$

For the dealer, the client's option is secret until it is necessary to make it public. The client's $<option>_{skc}$ is encrypted with TTP's public key PKT and has the signature of the client. If necessary, TTP can open it.

(b) The client prepares payment slip

$$S_C = <CStatus>_{skc}$$
$$CStatus = (GStatus,\ \mathcal{CC},\ D,\ \$AC,\ tcb_c)$$

where $\mathcal{CC}$ is the credit information of the client, D is the identification of the dealer, $\$AC$ is the quantity of money, and tcb_c is timestamp.

Encrypted Payment slip:

$$C_{SC} = P_{encr}(PKT, S_C).$$

The payment slip C_{SC} contains the information on game status, credit and betting. Using the techniques of $PEDLDLL$ discussed in the second section of this paper, $C_{SC}Cert$ could be constructed for checking that C_{SC} is the encrypted payment slip.

4. The dealer prepares his payment slip based on client's betting,

$$S_D = <DStatus>_{skd}$$
$$DStatus = (CStatus,\ \mathcal{CD},\ \$AD,\ tdb_c)$$

where $\mathcal{CD}$: credit information of the dealer; $\$AD$: quantity of money; tdb_c: timestamp.

Encrypted payment slip:

$$C_{SD} = P_{encr}(PKT, S_D)$$

The payment slip C_{SD} contains the current information of game status, credit and betting. Using the techniques of $PEDLDLL$, $C_{SD}Cert$ is constructed for checking that C_{SD} is the encrypted payment slip.

5. The client sends his option to the dealer by message

$$P_{encr}(PKD,\ option),\ \ <H(option)>_{skc}$$

The message is encrypted with the dealer's public key PKD. The dealer can read the message and know client's option. Based on client's options and G_{ID}, the dealer can get the result of the game.

6. The dealer sends the client token $P_{encr}(PKC,\ G_{ID},\ result)$ and hence, the client can get G_{ID}. At this time, the client knows both the G_{ID} and his option, and hence he can run the game and get the result. If the client wins, the dealer also sends the client token $P_{encr}(PKC,\ C_{SD})$. The client gets the payment slip C_{SD}.

7. If the dealer wins, the client sends the dealer $P_{encr}(PKC,\ C_{SC})$ and the the dealer gets payment slip C_{SC}).

The whole process of the gambling and payment is fair in the above protocol. Before step 5, neither the dealer and the client can get the result of the game. The dealer has G_{ID} as secret, the client has *options* as secret. Both of them encrypt their secrets with TTP's public key at first. They make their secrets public in step 5 and step 6. If the loser refuses to pay, the winner can bring encrypted payment slip C_{SC} and C_{SD} to TTP. TTP can then open them and check the result of the game. C_{SC} contains the information on client's betting. C_{SD} contains the information on dealer's betting. Both of them are necessary for the TTP to check the betting process and the result. Based on checking the result, TTP can forward S_C to the dealer if the dealer wins. It can forward S_D to the client if the client wins. The whole process is fair for both the client and the dealer.

Before the client has sent his C_{SC}, the client has the right to quit the protocol. Before the dealer sends his C_{SD}, the dealer has the right to quit the protocol. In above cases, both of them do not have the other party's encrypted payment slip, so they can not get any payment or useful information of the game. The protocol is aborted but the process is fair. If one party has the other party's encrypted payment slip, he can bring both C_{SC} and C_{SD} to TTP. The protocol can finish with the help of TTP. TTP can get all information of the game and betting from C_{SC} and C_{SD}. TTP can get the result of the game and forward the payment to the winner.

The above protocol can be extended in a real application. In step 3, the client perhaps discloses part of his options for the game to progress or prepare encrypted payment slips for betting. In step 4, the dealer perhaps provides some information of the current game or prepare encrypted payment slip as a response to client's betting. In step 5, the client makes his current options public. Payments are given if the client chooses to trust the dealer. This kind of processes can repeat again and again until the end of the game or the client chooses not to trust the dealer. The dealer sends the client G_{ID}. The client can run the game on his local machine to check the whole process of gambling. If cheating occurs, he can bring all encrypted payment slips to TTP to prove that the dealer is cheating. In this case, the whole process is fair.

2.4 Pure Luck Games

Many casino games are solely games of chance [1]. This kind of games can be abstracted to generating some random number. We assume that there are only two parties in our pure luck game. They will cooperate with each other to generate the random number and they will bet on the result of the random number. In this part, we will discuss the fairness both of game process and the dual-payment between the two parties.

The following is the two-party protocol for generating a random number and how they bet and arrange payment on the output of the random number. We outline the protocol as follows:

1. Alice generates a random number and signs the hash of the random number, Alice sends Bob the following:

$$M_1 = H(R_A),\ \ <H(R_A)>_{ska},\ P_{encr}(PKT,\\ <R_A>_{ska})$$

$P_{encr}(PKT,\ \ <R_A>_{ska})$ is the TTP's public key encryption of the random number R_A with the signature of Alice.

2. Bob generates a random number and signs the hash of the random number, Bob sends Alice the following:

$$M_2 = H(R_B), <M_1, H(R_B)>_{skb},\\ P_{encr}(PKT,\ \ <R_B>_{skb})$$

$P_{encr}(PKT,\ \ <R_B>_{skb})$ is the TTP's public key encryption of the random number R_B with the signature of Bob.

3. Alice prepares her encrypted payment slip and the certificate of the encrypted payment slip

$$S_A = <A, \mathcal{CA}, B, M_1, M_2, Abetting, t_a>_{ska}$$

where A is Alice's identification; $\mathcal{CA}$ is credit information (defined in section 3.1) ; B is Bob identification; M_1 and M_2 are messages of step 1 and step 2. *Abetting* contains Alice's betting options and amount of money for this betting. t_a is timestamp.

Encrypted Payment slip is:

$$C_{SA} = P_{encr}(PKT, S_A)$$

With the technique of $PEDLDLL$, Alice constructs certificate of the encrypted payment slip $C_{SA}Cert$.

Alice sends Bob

$$M_3 = (ABetting, C_{SA}, C_{SA}Cert\ )$$

4. Bob prepares his encrypted payment slip and the certificate of the encrypted payment slip

$$S_B =< B, \mathcal{CB}, A, M_3, Bbetting, t_b >_{skb}$$

where B is Bob's identification; $\mathcal{CB}$ is credit information (defined in section 3.1) ; A is Alice's identification; M_3 is the message of last step. *Bbetting* contains Bob's response of Alice's betting which contains Bob's amount of money on this betting. t_b is timestamp.

Encrypted Payment slip is:

$$C_{SB} = P_{encr}(PKT, S_B)$$

Using the technique of *PEDLDLL*, Bob constructs certificate of the encrypted payment slip $C_{SB}Cert$.

Bob sends Alice

$$M_4 = Bbetting, C_{SB}, C_{SB}Cert$$

5. Alice sends Bob the actual value of number R_A:

$$M_5 = R_A, < M_4, R_A >_{ska}$$

6. Bob sends Alice the actual value of number R_B:

$$M_6 = R_B, < M_5, R_B >_{skb}$$

7. Both Alice and Bob computes the random number

$$R = R_A \ XOR \ R_B$$

8. If Alice loses, Alice sends Bob her payment slip S_A; if Bob loses, Bob sends Alice his payment slip S_B.

In this protocol, Alice and Bob send their hashed digest at first, then they give their betting and prepare their payment slips. They encrypt their payment slips with TTP's public key and construct certificates for verifying the encrypted payment slips. Then they send their betting, encrypted payment slips and certificates for verifying their encrypted payment slips. In this step, Alice and Bob have given their betting with payments and cannot change; but they cannot get the money at this time because payment slips are encrypted with TTP's public key. In steps 5 and 6, they send their actual chosen values to the other party and then both Alice and Bob can compute the number.

The protocol is fair for both Alice and Bob. If the loser refuses to pay, the winner could bring encrypted payment slips C_{SA} and C_{SB} to TTP. TTP can open them and check the process of betting. TTP can then forward the payment slip to the winner. Alice can quit the protocol at or before step 3, Bob can quit the protocol at or before step 4. At any other time, if one party stop the protocol, the peer party can bring C_{SA} and C_{SB} to TTP, and the protocol can continue to the end. The winner receives the payment with the help of TTP.

3 Efficient Fair Mental Poker Protocol

3.1 Typical Former Protocols of Mental Poker

3.1.1 Protocol Based on Individual Card Cryptosystem

Adi Shamir, Ronald Rivest, and Leonard Adleman [12] utilized commutative cryptosystems to develop their mental poker protocol. Let E_A and D_A be Alice's encryption and decryption functions, E_B and D_B be Bob's encryption and decryption functions respectively. In real implementation, Alice and Bob agree on a large prime number p, and respectively choose secret keys $k = A$ and $k = B$, where $gcd(A, p-1) = gcd(B, p-1) = 1$. Then $E_k(x) \equiv x^k (\ mod\ p)$ and $D_k(x) \equiv x^z (\ mod\ p)$, where $kz \equiv 1 (\ mod\ p-1)$. The above cryptosystem is a commutative cryptosystem. For all message x, $E_A(D_B(x)) = D_B(E_A(x))$, $E_B(D_A(x)) = D_A(E_B(x))$, $E_A(E_B(x)) = E_B(E_A(x))$, $D_A(D_B(x)) = D_B(D_A(x))$. Alice and Bob will play the game as follows:

1. A deck of cards $\{1, ..., 52\}$ is used in the cryptosystem. Alice encrypts each card in the deck separately. Alice sends the set $\{E_A(1), ..., E_A(52)\}$ in a random order 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 could 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, and sends them back to Alice as 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 could decrypt and get {3, 11, 19, 23, 41}.
5. At the end of the game, they could exchange their encryption keys and verify that each played fairly.

Lipton [13] observed that the above implementation leaks at least one bit of information. 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 which cards are quadratic residues, and comparing them with encrypted cards, players could have a bit of information per card. Lipton provided some suggestions for the one bit information leak, but there is no guarantee that the result is secure [17].

3.1.2 Protocol Based on Permutation Cryptosystem

There are a series protocols [1][15][19] which are based on the multiple permutations. In the following, we will describe a popular protocol. There are three players Alice, Bob and Charles and one card salesman. They use the following steps to prepare a deck of cards:

1. Card salesman chooses a permutation π
2. Alice chooses three permutations A_a, A_b and A_c. Bob chooses three permutations B_a, B_b and B_c. Charles chooses three permutations C_a, C_b and C_c. All the above permutations are sent to the card salesman confidentially (only the sender and the card salesman know them).
3. Card salesman computes and broadcasts $\delta_a = B_a^{-1}C_a^{-1}A_a^{-1}\pi^{-1}$, $\delta_b = C_b^{-1}A_b^{-1}B_b^{-1}\pi^{-1}$, and $\delta_c = A_c^{-1}B_c^{-1}C_c^{-1}\pi^{-1}$.

If a player, for example Alice, wants to draw a card, the following protocol is used

1. Alice chooses $y = \pi(x)$ which is not in any player's hand and broadcasts y and $\delta_a(y)$.
2. Bob computes and broadcasts $B_a(\delta_a(y))$.
3. Charles computes and broadcasts $C_a(B_a(\delta_a(y)))$.
4. Alice computes $x = A_a(C_a(B_a(\delta_a(y))))$.
5. All players record that $y = \pi(x)$ has been in Alice's hand.

At the end, all permutations are published to check the fairness of the game. The above protocol could guarantee that a player can draw a card which is not in anyone's hand and only he could know what the card is. If the card salesman and at least one player plays fairly, there is no way for a player or group of colluding players to get information of cards not in their own hands. This protocol requires a card salesman to choose the random π and broadcast permutations. If the card game is used for gambling, the assumption that the card salesman be fully trusted is not a good one. Another aspect of this this permutation based poker scheme is that cheating can only be detected at the end of the game and not during the protocol run.

3.2 Multi-Party Encryption and Decryption

Here 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 [32].

1. Encryption:

   The original message is x. A chooses random number r_A, and the result of encryption with $\mathcal{K}_A$ has two parts y_{1A} and y_{2A}:

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

   B chooses random number r_B and encrypts the ciphertext of A's encryption (actually B encrypts y_{2A}) 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 ciphertex y_{1A}, y_{1B}, y_{2AB}.

2. Decryption:

   If A uses his private key to decrypt first,

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

   and then B uses his private key to decrypt

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

   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. If a different order is used for encryption, the final cipher-text is the same. If a different order is used for decryption, the original message could be obtained as well. The order of the encryptions and decryptions will not change the result. The mental poker protocols described in this paper are based on the power of the above commutative cryptosystems.

3.3 Our Mental Poker Protocols

Our target is to design mental poker protocols for multiple players to play fair on-line mental poker games. The mental poker protocols must provide fairness for the involved parties. The fair mental poker protocols 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 mental poker protocols presented in this paper, there is not a trusted third party involved during the game. These protocols focus on the processes of shuffling and dealing the cards only and is suitable for any set of cards. Our protocols deal with cards one by one which is different from the protocols based on permutations of cards [15]. 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.

3.3.1 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.

3.3.2 Protocol Description

In this section, we propose two mental poker protocols based on the cryptosystem with multiple encryptions and decryptions described in sub section 3.2. 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.

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.

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.

3.4 Discussion of Our Mental Poker Protocols

Here we discuss some important security properties of mental poker protocols described in this paper. We compared our protocols with previously published protocols.

(I) 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 [10][12]. Shamir et al mental poker is efficient but there is at least one bit information leak [13][17]. To avoid the information leak, Zhao et al [28] proposed an efficient TTP-free mental poker based on ElGamal cryptosystem. Unfortunately, a security flaw is introduced [33]. The mental poker protocols provided in this paper are based on our previous research [29], the security flaw has been removed and the complete confidentiality of cards has been achieved. Actually, the mentioned security flaw comes from the reusing of the same random number in the encryptions of the whole set of cards. When different random numbers are used in the encryptions of the set of cards, the mentioned security flaw is removed.

(II) Without card salesman

There is a card salesman involved in previous protocols [1][15][19] that are based on multiple permutations. The fairness of this kind of protocols is based on the assumption that the card salesman is fully trusted. In real gambling, such an assumption is definitely not a good one. We can not assume the existence of such a fully trusted party in online gambling. The mental poker protocols presented in this paper get rid of the card salesman completely.

(III) Any Number of Players

Based on the commutativity of multi encryptions and decryptions, it is convenient to expand the protocols to multi-players. With the same prime number p, every player, for example X, has key pair $\mathcal{K}_X = \{(p, \alpha_X, a_X, \beta_X) : \beta_X \equiv \alpha_X{}^{a_X} (\bmod\ p)\}$. In the card shuffling process, every player X chooses a set of secret random numbers $\{r_{X1}, r_{X2}, ..., r_{X52}\}$. All cards are multi-encrypted by all players. In the card dealing, when player X draws a card, all other players decrypt the card, and only player X can open the card. All players delete the card from the available set.

(IV) Security Against Player Collusions

The presented mental poker protocols can guarantee the minimal effect of collusion. Even if two players collude, they can only obtain each other's cards but not a card of a third player. Because every card is multi-encrypted by all the players, a card is opened only in the case that all players have decrypted it. Any subset of players can not know anything about the cards of other players. No collusion among cheating players can affect the cards drawn by an honest player and untouched cards.

(V) Complete Confidentiality of Strategy

The protocols presented in this paper asks players to reveal all information at the end of the game. It makes it impossible for the players to bluff. Real poker players would never accept to play such a game. Fortunately, if a dealer is involved, it is very easy to modify the above protocols. When shuffling cards, every player X chooses his secret random number set $\{r_{X1}, r_{X2}, ..., r_{X52}\}$ and sends $\{< H(r_{X1}) >, < H(r_{X2}) >, ..., < H(r_{X52}) >\}$ to the dealer. During the game, every player sends the information of his actions (for retrieving in the future, except opened cards) to the dealer. At the end of the game, every player sends his secret random number set to the dealer. The dealer is able to check the fairness of the whole game. During the game, the card information is confidential to the dealer. The dealer is the only person who can know the strategy of each player at the end of the game. Such an assumption is reasonable and acceptable. It is much better than the assumption of a card salesman who is fully trusted and knows all card information during the game.

(VI) Efficiency and Clarity

The cryptosystem used in this paper is based on the popular ElGamal cryptosystem. For a game of two players, there are only 104 times ElGamal encryptions and decryptions. For a game of n players, there are $52 \times n$ ElGamal encryptions and $52 \times n$ ElGamal decryptions. The protocol is very efficient. For a group of players, after the system has been setup, they can use their encryption/decryption key pairs and public/private key pairs for

multiple games. For a new game, the players only need to choose new secret random numbers (encryption parameters). There are several other successful protocols based on zero-knowledge proofs. Unfortunately they are not practical and are often very complicated and messy. They need a fairly long computation time to shuffle a deck of cards.

4 Conclusions

In the fair online gambling scheme, we have presented protocols for games from authority organization and pure luck games which are useful in implementing some real online games. The protocols can guarantee the fairness of both the games and payments. The major feature of these protocols lies in the use of encrypted payment slips and certificates of the encrypted payment slips.

The presented mental poker protocols have achieved the major requirements of a complete poker system. The protocols are secure, efficient and suitable for any number of players. The presented mental poker protocols get rid of the card salesman entity completely and there is minimal effect due to collusion of players. With the introduction of a dealer, the strategies of players can be made confidential to other people (except the dealer). In this case, the dealer only becomes aware of the strategies of players at the end of the game.

For online gambling, there are a fairly large number of card games. The gambling requires actions such as placing bets and dealing with payments. The presented mental poker protocols are based on individual cards. It is easy to combine this kind of protocols with the management protocols of whole gambling processes. Based on the fair online gambling scheme and fair mental poker protocols, the fairness required to use card games in online gambling can be achieved.

Our fair online gambling scheme is only suitable for some online games. There are many other open problems related to the fairness of online gambling games; for example, if there are more than two people involved in a game, how to deal with the issue of collusion? The fair mental poker protocols have some open problems as well, such as how to return a card to the deck. There are still many open issues in designing fair, secure and efficient protocols for different kinds of online games for the purpose of gambling.

References

[1] C. Hall and B. Schneier. “Remote Electronic Gambling”. In Proceedings of 13th Annual Computer Security Applications Conference, ACM Press IEEE, pp. 227-230, 1997.

[2] R. Oppliger and J. L. Nottaris. “Online casino”. At URL: http://citeseer.nj.nec.com/126635.html.

[3] J. Garcia, F. Cuppens, F. Autrel, J. Castella-Roca, J. Borrell, G. Navarro and J.A. Ortega-Ruiz. “Protecting on-line casinos against fraudulent player drop-out”. In Proceedings of Information Technology: Coding and Computing, Volume 1, pp. 500-505, 2005.

[4] R. Hauser, M. Steiner and M. Waidner. “Micro-payments based on iKP”. Technical Report No. 89269, 1996.

[5] “SET Secure Electronic Transaction 1.0”. Technical Report (May 1997). Mastercard.

[6] P. Janson and M. Waidner. “Electronic payment systems”. Semper/IBM Zurich Research Lab. Activity Paper 211ZR018, May 1996.

[7] N. Asokan, M. Steiner, and M. Waidner. “The state of the art in electronic payment systems”. IEEE Computer, pp. 28- 35, 1997.

[8] Y. Mu and V. Varadharajan 1998. “A new scheme of credit based payment for electronic commerce”. In Proceedings of 23rd Local Area Networks Conference, Boston, IEEE Computer Society, pp. 278-284, 1998.

[9] F. Fitzek, G. Schulte, and M. Reisslein. “System architecture for billing of multi-player games in a wireless environment using GSM/UMTS and WLAN Services”. In Proceedings of the 1st Workshop on Network and System Support for Games (NetGames), Braunschweig, Germany, pp. 58-64, 2002.

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

[11] W. Zhao, V. Varadharajan and Y. Mu. “Fair Online Gambling”. In Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000), pp. 394-400, 2000.

[12] A. Shamir, R. Rivest and L. Adleman. “Mental Poker”. Mathematical Gardner (D.E. Klarner, ed.), Wadsworth International, pp. 37-43, 1981.

[13] R. Lipton. “How to Cheat at Mental Poker”. In Proceedings of the AMS Short Course in Cryptography, 1981.

[14] 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, pp. 270-299, 1982.

[15] S. Fortune and M. Merritt. “Poker Protocols”. Advances in Cryptology - Proceeding of CRYPTO 84, Springer-Verlag, pp. 454-464, 1985.

[16] M. Yung. “Cryptoprotocols: Subscriptions to a Public Key, the Secret Blocking, and the Multu-Player Mental Poker Game”. Advances in Cryptology - CRYPTO’84 Proceedings, Springer-Verlag, pp. 439-453, 1985.

[17] D. Coppersmith. “Cheating at Mental Poker. Advances in Cryptology - CRYPTO’85 Proceedings, Springer-Verlag, pp. 104-107, 1986.

[18] K. Kurosawa, Y. Katayama, W. Ogata and S. Tsujii. “General Public Key Reside Cryptosystems and Mental Poker Protocols”. Advances in Cryptology - CRYPTO’90 Proceedings, Springer-Verlag, pp. 374-388, 1991.

[19] I. Barany and Z. Furedi. “Mental Poker with Three or More Players”. Technical Report, Mathematical Institute of the Hungarian Academy of Science, 1983.

[20] C. Crepeau. “A Secure Poker Protocol that Minimizes the Effect of Player Coalitions”. Advances in Cryptology - CRYPTO’85 Proceedings, Springer-Verlag, pp. 73-86, 1986.

[21] C. Crepeau. “A Zero-Knowledge Poker Protocol that Achieves Confidentiality of the Players’ Strategy or How to Achieve an Electronic Poker Face”. Advances in Cryptology - CRYPTO’86 Proceedings, Springer-Verlag, pp. 239-247, 1987.

[22] C. Crepeau and J. Killian. “Discrete Solitary Games. Advances in Cryptology - CRYPTO’93 Proceedings, Springer-Verlag, pp. 319-330, 1994.

[23] Keen, P. et al. “Electronic commerce relationships: trust by design”. Preintice-Hall, New Jersey, 2000.

[24] S. Even, O.Goldreich and A. Lempel. “A randomized protocol for signing contracts”. CACM, Vol.28, No.6, pp. 637-647, 1985.

[25] J. Zhao and D. Gollmann. “An efficient non-repudiation protocol”. In Proceeding of 10th IEEE Computer Security Foundations Workshop, Rockport, Massachusetts, pp. 126-132, 1997.

[26] N. Asokan, M. Schunter and M. Waidner. “Optimistic protocols for fair exchange”. In Proceedings of 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, pp. 6-17, 1997.

[27] F. Bao, R. Deng and W. Mao. “Efficient and Practical Fair Exchange Protocols with Off-line TTP”. In Proceeding of 1998 IEEE Symposium on Security and Privacy, pp. 77-85, 1998.

[28] W. Zhao, V. Varadharajan and Y. Mu. "A secure mental poker protocol over the internet". In Proceedings of Australian Information Security Workshop, Adelaide, Australia. Australian Computer Society, Volume 21, pp. 105-109, 2003.

[29] W. Zhao and V. Varadharajan. "Efficient TTP-free mental poker protocols". In Proceedings of Information Technology: Coding and Computing, Volume 1, pp. 745-750, 2005.

[30] F. C. Gartner, H. Pagnia and H. Vogt. "Approaching a formal definition of fairness in electronic commerce". In Proceedings of the International Workshop on Electronic Commerce (WELCOM'99), Lausanne, Switzerland, 1999.

[31] V. Shmatikov and J. C. Mitchell. "Analysis of a fair exchange protocol". In Proceedings of the 1999 FLoC Workshop on Formal Methods and Security Protocols, 1999.

[32] T. ElGamal. "A public-key cryptosystem and a signature scheme based on discrete logarithms". IEEE Transaction on Information Theory, Volume 31, pp. 469-472, 1985.

[33] J. Castella-Roca and J. Domingo-Ferrer. "On the security of an efficient ttp-free mental poker protocol". In Proceedings of Information Technology: Coding and Computing, Volume 2, pp. 781-784, 2004.

[34] D.Chaum and T.P.Pedersen. "Wallet databases with observers". Advances in Cryptology -CRYPTO'92 Proceedings, Springer-Verlag, pp. 89-105, 1992.

[35] E. R. Verheul and H. C. A. van Tilborg. "Binding Elgamal: A fraud-detectable alternative to key-escrow proposals". Advances in Cryptology - EUROCRYPTO' 97 Proceedings, Springer-Verlag, pp. 119-133, 1997.

[36] M. Stadler. "Publicly verifiable secret sharing". In Proceeding of Eurocrypto' 96, LNCS 1070, Springer-Verlag, pp. 190-199, 1996.

Author Biographies

Weiliang Zhao He is currently a Ph.D candidate in School of Computing Information Technology of University of Western Sydney. He got Master of Honors - computing and information technology from University of Western Sydney in 2003. He got Bachelor of Science - physics from Peking University in 1988. His current areas of research interest include E-commerce security, security for Internet applications and Web Services and models and architectures of trust management.

Vijay Varadharajan He is currently the Microsoft Chair and Professor of Computing at Macquarie University. He received his Ph.D in Computer and Communication Security in the U.K in 1984, sponsored by BT Research Labs. He received a Bachelor degree in Electronic Engineering from Sussex University in 1981. His current areas of research interest include security in high speed networks, security for large distributed systems, security policies and management in distributed applications, Internet security, secure electronic commerce and payment models, secure mobile agents, wireless security, security models and architectures and security protocols.

Appendix

Appendix A: Equality Proof of Knowledge

The scheme of equality proof of knowledge was initially proposed by Chaum and Pedersen [34] and Verheul and Tilborg [35]. The scheme is about proving knowledge of something without revealing anything about its content. The public information includes a prime number p and a generator $g_i \in Z_p^*, \ \ i = 1, 2, \cdots, l$, where l is the confidence level. In order to prove x, the prover chooses $r \in Z_p^*$ and computes

$$a_i = g_i^r \mod p,$$
$$h_i = g_i^x \mod p.$$

Challenges c and z are calculated as follows

$$c = H(g_1||g_2||\cdots||g_l||a_1||a_2||\cdots||a_l||h_1||h_2||\cdots||h_l),$$
$$z = cx + r \mod p.$$

The verifier will check the following equation to prove the knowledge

$$g_i^z \stackrel{?}{=} h_i^c a_i \mod p.$$

For all i, $g_i^z = h_i^c a_i \mod p$ indicates that the prover has the knowledge; otherwise, he does not.

Appendix B: Proof of Equivalence of Discrete Logarithm to Discrete Log-logarithm(PEDLDLL)

PEDLDLL was initially proposed by Stadler [36]. For two given primes p and q (where $p = 2q + 1$), let $x, y, z \in Z_q^*$ and $X, Y \in Z_p^*$. There exists an $\alpha \in \{1, 2, \cdots q - 2\}$ such that $y = x^\alpha \mod q$ and $Y = X^{z^\alpha} \mod p$. Without revealing α and z^α, a prover, who knows α, can generate a certificate to prove that $y = x^\alpha \mod q$ and $Y = X^{z^\alpha} \mod p$.

If the confidence level is l, for $i = 1, 2, \cdots, l$, the prover chooses $w_i \in \{1, 2, \cdots, q - 2\}$ and computes $t(x_i) = x^{w_i} \mod q$, $t(X_i) = X^{z^{w_i}} \mod p$. Then he could get

$$c = H_l(x||y||z||X||Y||t(x_1)||t(X_1)||\cdots||t(x_l)||t(X_l)).$$

For every bit $c = c_1 c_2 \cdots c_l$, the prover computes $R = (r_1, r_2, \cdots, r_l)$, where $r_i = w_i - c_i \alpha \mod q - 1$. The certificate is given by (R, c).

During certificate verification, the verifier will check whether

$$c = H_l(x||y||z||X||Y||u_1||U_1||\cdots||u_l||U_l)$$

where $u_i = x^{r_i} y^{c_i} \mod q$ and

$$U_i = \begin{cases} X^{z^{r_i}} \mod p & if \ \ c_i = 0 \\ Y^{z^{r_i}} \mod p & if \ \ c_i = 1 \end{cases}$$