Dropout-Tolerant TTP-Free Mental Poker

Jordi Castellà-Roca, Francesc Sebé, and Josep Domingo-Ferrer

Rovira i Virgili University of Tarragona,
Dept. of Computer Engineering and Maths,
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{jordi.castella, francesc.sebe, josep.domingo}@urv.net

Abstract. There is a broad literature on distributed card games over communications networks, collectively known as *mental poker*. Like in any distributed protocol, avoiding the need for a Trusted Third Party (TTP) in mental poker is highly desirable, because really trusted TTPs are not always available and seldom free. This paper deals with the player dropout problem in mental poker without a TTP. A solution based on zero-knowledge proofs is proposed. While staying TTP-free, our proposal allows the game to continue after player dropout.

Keywords: Mental poker, player dropout.

1 Introduction

According to Merryll Lynch, the online gambling business is expected to grow to $48 billion by 2010 and $177 billion by 2015. This booming turnover must be accompanied by enough security guarantees for online players; unfortunately, this is not always the case, especially as far as e-poker (mental poker) is concerned. Mental poker is played like a conventional card game with the difference that players communicate over a network and do not need to be in the same physical place. In this situation, cheating becomes especially tempting and must be prevented. A mental poker solution must offer all protocols needed to complete a game. These are: shuffling, drawing, discarding and opening.

The above protocols should offer the same security properties as conventional physical poker, plus some security properties specific to electronic gaming. Such properties were identified and enumerated by Crépeau in [4].

Dropout tolerance was not listed in [4] as a requirement, but it is nonetheless a major challenge in remote gaming. In electronic gaming, no one can prevent a player from quitting a game. Two kinds of dropout can be distinguished:

- **Intentional:** A player decides to quit the game. This may be attractive for a player to whom the game is not being favorable.
- **Accidental:** A player cannot go on playing, for example due to a network problem.

Whatever the reason for player dropout, the remaining players should be able to continue the game. If a Trusted Third Party (TTP) is controlling the game,

S. Katsikas, J. López, G. Pernul (Eds.): TrustBus 2005, LNCS 3592, pp. 30–40, 2005.

handling player dropout is greatly simplified. However, a TTP is not always available or desirable: it may not be trusted by everybody, it may charge some fee, etc. When no TTP is assumed, dropout becomes a nontrivial problem.

1.1 Contribution and Plan of This Paper

This paper proposes a solution for player dropout in mental poker without a TTP. The solution is based on zero-knowledge proofs and allows the game to continue after the dropout.

Section 2 reviews literature on TTP-free mental poker offering player confidentiality. Our proposed protocol is described in Section 3. Security is examined in Section 4. Finally, Section 5 is a conclusion. The Appendix contains the security proofs.

2 Background on TTP-Free Mental Poker Offering Player Confidentiality

All schemes mentioned in this section fulfill all security requirements identified in [4], including the confidentiality of player strategy. We next review them by focusing on their ability to handle player dropout.

Schemes [5,9] do not consider player dropout. In both proposals, each player has some secret information needed to draw cards from the deck. Without this information, the game cannot proceed.

In [1] it is proposed that players who quit the the game should disclose their secret information. However, this solution is only applicable if dropout is intentional *and* the player leaving the game is willing to collaborate. In case of accidental dropout (*e.g.* due to a network problem) or malicious intentional dropout, there is no guarantee that the remaining players can go on playing.

The schemes [7,10] represent each card in the deck by a different numerical value. During card shuffling, those values are encrypted and permuted by each player. The effect of encryption is analogous to reversing cards in a physical deck. A secret-sharing scheme is used, so that at least t players are needed to decrypt values. The goal is that the game can proceed if at least t players remain, which allows for some dropouts. In [7] the secret sharing scheme is applied to cards. Each value representing a card is divided into as many shares as there are players. Then each share is encrypted under the public key of a different player. A card cannot be decrypted unless at least t players co-operate. In [10], players create a key pair using the procedure proposed in [8]. Players generate a public key so that each player gets a share of the private key; thus, the private key cannot be used unless at least t players co-operate. Even if those schemes based on secret sharing do offer some dropout tolerance, the bad news is that secret sharing makes it possible for a sufficiently large collusion of players to obtain all deck information. Thus, dropout tolerance is traded off against collusion tolerance. This is frustrating because collusion tolerance is a basic security property already identified as relevant in [4].

3 Our Proposal

There is a first round where cards are dealt as in the poker game, and each player obtains five cards. If a player discards some cards from her hand, a new dealing round is started so that the player can obtain as many cards as she has discarded.

We use Protocol 1 to obtain a new deck of cards in each dealing round, in a similar way as proposed in [12]. In the second and successive dealing rounds each player *vetoes* (*i.e.* marks as unavailable) those cards that she has previously drawn. Protocol 2 is used to veto drawn cards. If a player obtains a vetoed card, she cannot use it and she does not know either the value of the vetoed card or who vetoed it; what the player can do is to show that she obtained a vetoed card and then draw a new card.

If a player leaves the game, the rest of players generate a new deck and use it in the game. The new deck includes the cards that were drawn by the player who left the game, because the latter is no longer there to veto her cards when the new deck is generated.

We shall use the following notation in the subsequent protocols.

n : number of players (we assume some ordering among the n players);
$\mathcal{P}_i$: the i-th player in the ordered set of n players;
λ_i : set of cards in $\mathcal{P}_i$'s hand;
Λ : set of all cards in the hands of all players, *i.e.* $\Lambda = \cup_{i=1}^{n} \lambda_i$;
δ_i : set of cards discarded by $\mathcal{P}_i$.

3.1 System Set-Up

Before a game starts, players $\mathcal{P}_1, \ldots, \mathcal{P}_n$ must set some parameters. They choose a large prime p so that $p = 2q+1$ and q is also prime; they also pick one element $g \in \mathbb{Z}_p^*$ of order q.

Using the key generation protocol described in [6], players jointly generate a public key $y = \prod_{i=1}^{n} y_i$. Each player $\mathcal{P}_i$ keeps her corresponding share α_i of the private key and publishes $y_i = g^{\alpha_i}$.

3.2 Deck Generation

Each card is represented by a value jointly computed by all players in Protocol 1. We first explain what Protocol 1 does and then describe the protocol in detail.

Let us assume that we are in the k-th dealing round. We can see in Step 1 of Procotol 1 below that every player uses Procedure 1 to compute 52 new values. These values are sent to the rest of players in Step 2 of Protocol 1. Once every player gets the new values from the rest of players, the new deck D_k is computed by all players at Step 3. We use the term face-up deck of cards because every player can see the value of each card; the j-th value $d_{k,j}$ in D_k represents the j-th card in the deck.

If $d_{k,j}$ is a face-up card, then we denote by $e_{k,j}$ the corresponding face-down card. $e_{k,j}$ contains the encrypted version of the exponents that have been used

to compute $d_{k,j} \in D_k$ from $d_{k-1,j} \in D_{k-1}$. To prove ownership of a card $d_{k,j}$, a player must prove knowledge of those exponents, *i.e.* prove knowledge of the discrete logarithm $\log_{d_{k-1,j}}(d_{k,j})$.

In the first round, all cards are available and $E_1 = C_{1,0}$ is the face-down deck of cards without shuffling (see Step 4 of Protocol 1). In subsequent rounds, each player vetoes the cards in her hand using Protocol 2 (called at Step 5a of Protocol 1); the goal is that cards already drawn should become unavailable. After using Protocol 2, a player gets one re-masking factor for each card in the deck; a vetoed card is re-masked with a factor which does not allow decryption, whereas a non-vetoed card is re-masked with a factor allowing decryption.

In Step 5b players re-mask the encrypted exponents with these factors, and obtain the face-down deck of cards without shuffling, $C_{k,0}$.

We denote by D_l the deck of cards of the l-th round; we denote by $\mathbf{D}$ the set of all decks that have been generated in all rounds, *i.e.* $\mathbf{D} = \{D_1, \ldots, D_k\}$. In order to run our protocol, we define $D_0 = \{d_{0,1}, \ldots, d_{0,52}\}$, where $d_{0,j} = g$, $\forall j \in \{1, \ldots, 52\}$.

Protocol 1 $(k \geq 1, D_{k-1})$

1. *Each player $\mathcal{P}_i$ uses Procedure 1 on D_{k-1} and obtains $D_{k,i} = \{d_{k,i,1}, \ldots, d_{k,i,52}\}$ and $E_{k,i} = \{e_{k,i,1}, \ldots, e_{k,i,52}\}$, where $d_{k,i,j} = d_{k-1,j}^{m_{k,i,j}}$ and $e_{k,i,j} = E_y(m_{k,i,j})$;*
2. *Each $\mathcal{P}_i$ publishes $D_{k,i}$ and $E_{k,i}$;*
3. *All players compute the face-up deck of cards $D_k = \{d_{k,1}, \ldots, d_{k,52}\}$ and $E_k = \{e_{k,1}, \ldots, e_{k,52}\}$, where $d_{k,j} = \prod_{i=1}^{n} d_{k,i,j} = d_{k-1,j}^{m_{k,1,j}+\ldots+m_{k,n,j}}$ and $e_{k,j} = \{e_{k,1,j}, \ldots, e_{k,n,j}\}$;*
4. *If $k = 1$, players compute the face-down deck of cards $C_{1,0} = \{c_{1,0,1}, \ldots, c_{1,0,52}\}$, where $c_{1,0,j} = e_{1,j} \in E_1$;*
5. *If $k > 1$ then players do the following*
   (a) *Run the vetoing protocol (Protocol 2) and obtain $G_k = \{g_{k,1}, \ldots, g_{k,52}\}$, where $g_{k,j} = \{g_{k,1,j}, \ldots, g_{k,n,j}\}$;*
   (b) *Compute the face-down deck of cards $C_{k,0} = \{c_{k,0,1}, \ldots, c_{k,0,52}\}$, where $c_{k,0,j} = e_{k,j} \cdot g_{k,j} = \{e_{k,1,j} \cdot g_{k,1,j}, \ldots, e_{k,n,j} \cdot g_{k,n,j}\}$. Drawn cards in this face-down deck have been vetoed.*

In the k-th dealing round, each player $\mathcal{P}_i$ computes a new value $d_{k,i,j}$ for each card j. This new value is obtained from $d_{k-1,j}$ (the value used in round $k-1$ to represent card j) raised to a random value $m_{k,i,j}$. The exponent $m_{k,i,j}$ is encrypted into $E_y(m_{k,i,j})$ and sent along with the new value. Players use Procedure 1 to compute these new values for each card.

Procedure 1 (y, p, D)

1. *For each d_j in $D = \{d_1, \ldots, d_{52}\}$ do:*
   (a) *generate a random value m_j, where $2 < m_j < q$;*
   (b) *compute $d_j^{m_j}$;*

   (c) *encrypt* m_j *into* $E_y(m_j)$ *under public key* y*;*
   (d) *prove in zero-knowledge to the rest of players that* $E_y(m_j)$ *is the encryption of* $\log_{d_j}(d_j^{m_j})$ *using [11];*
2. *Return the sets* $D' = \{d_1^{m_1}, \ldots, d_{52}^{m_{52}}\}$ *and* $E = \{E_y(m_1), \ldots, E_y(m_{52})\}$.

Prior to describing Protocol 2 we define $\xi_{l,i}$ as the number of cards that $\mathcal{P}_i$ has drawn in the l-th dealing round; we also define ξ_i as the sum of all cards drawn by $\mathcal{P}_i$ in all previous dealing rounds, that is, $\xi_i = \sum_{l=1}^{k-1} \xi_{i,l}$.

In Step 1a of Protocol 2 each player in turn computes a re-masking factor for each card $d_{k,j} \in D_k$. Using the construction of [3], $\mathcal{P}_i$ proves in Step 1b of Protocol 2 that $52 - \xi_i$ factors have been properly computed (as many factors as the number of cards $\mathcal{P}_i$ has not drawn); in this proof, $\mathcal{P}_i$ does not reveal which subset of factors was properly computed. In Step 1c, $\mathcal{P}_i$ again uses the construction of [3] to prove that she has computed ξ_i re-masking factors which veto the cards $\mathcal{P}_i$ has drawn (see in Section 4 the lemma that a player vetoes the cards she has drawn). The re-masking factors that veto all drawn cards by any player are pooled together in Step 2.

Protocol 2 (D_k)

1. *For each* $\mathcal{P}_i$ $(i = 1, \ldots, n)$*:*
   (a) $\mathcal{P}_i$ *uses Procedure 2 with* $(D_k, \lambda_i, \delta_i, \mathbf{D})$ *and obtains* $G_i = \{(u_{i,1}, v_{i,1}), \ldots, (u_{i,52}, v_{i,52})\}$.
   (b) $\mathcal{P}_i$ *proves in zero-knowledge that at least* $52-\xi_i$ *values* $(u_{i,j}, v_{i,j})$ *properly re-mask a card,* i.e. *they do not veto the card. This is done using the construction of [3] in order to show that* $\mathcal{P}_i$ *can correctly perform at least* $52-\xi_i$ *executions of the set of zero-knowledge proofs* $\{CP_{i,1}, \ldots, CP_{i,52}\}$*, where* $CP_{i,j} = CP(g, y, u_{i,j}, v_{i,j})$[1].
   (c) *For* $l = 1$ *to* k*,* $\mathcal{P}_i$ *proves in zero-knowledge that she has vetoed as many cards as the number* $\xi_{l,i}$ *of cards she obtained in the* l*-th dealing round. This is done using the construction of [3] in order to prove that she can perform at least* $\xi_{l,i}$ *executions among the following set of zero-knowledge proofs* $\{CP_{l,i,1}, \ldots, CP_{l,i,52}\}$*, where* $CP_{l,i,j} = CP(d_{l-1}, u_{i,j}, d_{l,j}, d_j)$*;*
2. *Compute* $G = \{g_1, \ldots, g_{52}\}$*, where* $g_j = (u_j, v_j)$*, and* $u_j = \prod_{i=1}^{n} u_{i,j}$ *and* $v_j = \prod_{i=1}^{n} v_{i,j}$*, with* $(u_{i,j}, v_{i,j}) \in G_i$*;*
3. *Let* $G_0 = \{(g_{0,1,1}, \ldots, g_{0,n,1}), \ldots, (g_{0,1,52}, \ldots, g_{0,n,52})\} := G$*, that is,* $g_{0,\zeta,j} = g_j$ $\forall \zeta \in \{1, \ldots, n\}$ *and* $\forall j \in \{1, \ldots, 52\}$*, and* $g_j = (u_j, v_j) \in G$*;*
4. *For each* $\mathcal{P}_i$ $(i = 1, \ldots, n)$*:*
   (a) *receive* $G_{i-1} = \{(g_{i-1,1,1}, \ldots, g_{i-1,n,1}), \ldots, (g_{i-1,1,52}, \ldots, g_{i-1,n,52})\}$ *from* $\mathcal{P}_{i-1}$*;*
   (b) *compute* $G_i = \{(g_{i,1,1}, \ldots, g_{i,n,1}), \ldots, (g_{i,1,52}, \ldots, g_{i,n,52})\}$*, where* $g_{i,\zeta,j} = g_{i-1,\zeta,j}^{r_{i,\zeta,j}} = (u_{i-1,\zeta,j}^{r_{i,\zeta,j}}, v_{i-1,\zeta,j}^{r_{i,\zeta,j}})$ *and* $1 < r_{i,\zeta,j} < q$ *is a value obtained at random;*

[1] We will denote by $CP(g, y, u, v)$ the Chaum-Pedersen [2] zero-knowledge proof, *i.e.* the proof that $\log_g u = \log_y v$.

(c) *for each* $g_{i,\zeta,j} = (u_{i,\zeta,j}, v_{i,\zeta,j})$ *in* G_i *run* $CP(u_{i-1,\zeta,j}, v_{i-1,\zeta,j}, u_{i,\zeta,j}, v_{i,\zeta,j})$;
(d) *send* G_i *to the next player;*
5. *Return* G_n.

Procedure 2 is used by every player to compute the values used in re-masking. If card j is not vetoed then a pair (u_j, v_j) is computed such that $\log_g u_j = \log_y v_j$. However, if card j must be vetoed, a pair (u_j, v_j) such that $\log_g u \neq \log_y v$ is computed, in order to prevent a correct decryption.

As will be described in Section 3.4, when $\mathcal{P}_i$ obtains a card j at round k, $\mathcal{P}_i$ obtains the discrete logarithm $\tau_{k,j} = \log_{d_{k-1,j}} d_{k,j}$. In subsequent rounds $\mathcal{P}_i$ uses that logarithm to veto this card.

Procedure 2 $(D, \lambda, \delta, \mathbf{D})$

1. *For each* d_j *in* $D = \{d_1, \ldots, d_{52}\}$ *do:*
   (a) *if the card represented by* d_j *is in* $\lambda \cup \delta$ *do:*
      i. *generate a random value* R_j, *where* $1 < R_j < p$;
      ii. *let us assume that the card represented by* d_j *has been obtained in round* l. *In this case* $\tau_{l,j} = \log_{d_{l-1,j}}(d_{l,j})$ *is known (where* $d_{l-1,j}$ *and* $d_{l,j}$ *are in* $\mathbf{D}$*), so* $g_j = (u_j, v_j)$ *is computed, where* $u_j = d_j^{\tau_{l,j}^{-1}}$ *and* $v_j = R_j$;
   (b) *if the card represented by* d_j *is not in* $\lambda \cup \delta$ *do:*
      i. *generate a random* r_j, *where* $1 < r_j < q$;
      ii. *compute* $g_j = (u_j, v_j)$, *where* $u_j = g^{r_j}$ *and* $v_j = y^{r_j}$;
2. *Return* $G = \{g_1, \ldots, g_{52}\} = \{(u_1, v_1), \ldots, (u_{52}, v_{52})\}$.

3.3 Card Shuffling

This is done using the procedure described in [1]. The different players in turn shuffle and re-mask the face-down deck C_0 obtained with Protocol 1.

Protocol 3 (C_0)

1. *For each player* $\mathcal{P}_i$ $(i = 1, \ldots, n)$ *do:*
   (a) *Generate a permutation* σ_i *of 52 elements;*
   (b) *Permute the elements* $c_{i-1,j}$ *of the face-down deck* C_{i-1} *with* σ_i *to obtain* C_i^*;
   (c) *Re-mask the encrypted messages contained in each card of* C_i^* *without modifying their content to obtain* C_i; *this is done by re-masking all ciphertexts contained in each face-down card* $C_i^* = \{c_{i,1}^*, \ldots, c_{i,52}^*\}$, *where* $c_{i,j}^* = \{e_{i,j,1}^*, \ldots, e_{i,j,n}^*\}$; *specifically,* $\mathcal{P}_i$ *computes* $C_i = \{c_{i,1}, \ldots, c_{i,52}\}$, *where* $c_{i,j} = \{e_{i,j,1}^* \cdot E_y(1, r_{i,j,1}), \ldots, e_{i,j,n}^* \cdot E_y(1, r_{i,j,n})\}$; *values* $\{r_{i,j,1}, \ldots, r_{i,j,n}\}$ *are obtained at random;*
   (d) *Use the proof in [1] to prove in zero-knowledge that* C_i *is a permuted and re-masked version of* C_{i-1}.

After running the shuffling protocol, players get the set C_n, that is, the shuffled face-down deck of cards.

3.4 Card Drawing

The card extraction procedure is as follows. Let us assume that extraction is performed by player $\mathcal{P}_i$:

Protocol 4

1. *$\mathcal{P}_i$ randomly selects an element from C_n, namely, $c_j = (e_{j,1}, \ldots, e_{j,n})$;*
2. *$\mathcal{P}_i$ asks the rest of players to verifiably send her information to decrypt the messages $e_{j,\zeta}$ contained in c_j using [6];*
3. *After decrypting these messages, $\mathcal{P}_i$ obtains $\{m_1, \ldots, m_n\}$, where $m_\zeta = D(e_{j,\zeta})$;*
4. *$\mathcal{P}_i$ searches for an element $d_{k,t} \in D_k$ (the k-th dealing round deck) such that $d_{k-1,t}^{m_1+\ldots+m_n} \equiv d_{k,t}$;*
5. *If $d_{k,t} \in D_k$ then $\mathcal{P}_i$ stores $\tau_t = (m_1 + \ldots + m_n)$;*
6. *If $d_{k,t} \notin D_k$ then $\mathcal{P}_i$ has obtained a vetoed card. In this case, she shows that the card was vetoed and requests a new one.*

3.5 Card Opening

A player must prove to the rest of players that she is the owner of her cards. The card opening protocol is used to that end.

Let us assume that a player $\mathcal{P}_i$ has drawn a card $c_j \in C_n$. $\mathcal{P}_i$ has received the partial decryption from the rest of players, and she has verified that each partial decryption is correct.

$\mathcal{P}_i$ opens a card when she publishes the remaining part of the decryption of c_j. With this information, all players can know the value of c_j, that is, the decrypted card exponents $D(e_{j,1}), \cdots, D(e_{j,n})$. The decryption is verifiably performed as detailed in [6].

3.6 Card Discarding

A player discards a card when she commits herself to not using it. Let us assume that player $\mathcal{P}_i$ has drawn a card $c_j \in C_n$ using Protocol 4.

$\mathcal{P}_i$ discards c_j by sending a message *discard* with c_j to the rest of players. c_j is added to the set λ_i. If $\mathcal{P}_i$ wants to open a discarded card, the rest of players can detect the cheating because the card is in λ_i.

3.7 Player Dropout

In case one of the players leaves the game, the rest of players can go on playing. Assuming that player $\mathcal{P}_i$ with public key y_i leaves the game, the game public key is updated as $y := y/y_i$. Next, the rest of players continue as if player $\mathcal{P}_i$ had never joined the game. This implies that cards once extracted by $\mathcal{P}_i$ will be back in the deck.

4 Security

Security results in this section basically state that: i) vetoed cards cannot be opened; and ii) the set of vetoed cards is the set of drawn cards. Proofs are given in the Appendix.

Lemma 1. *If $\mathcal{P}_i$ succeeds in performing $CP_{i,j} = CP(g, y, u_{i,j}, v_{i,j})$ at Step 1b of Protocol 2, then $(u_{i,j}, v_{i,j})$ will not veto the face-down card $c_{k,0,j}$ at Step 5b of Protocol 1.*

The Corollary below follows from the above lemma and from Step 1b of Protocol 2.

Corollary 1. *Let 52 be the total number of cards. Let ξ_i be the cards drawn by a player $\mathcal{P}_i$. Then the number of cards x_i not vetoed by $\mathcal{P}_i$ is such that $x_i \geq 52 - \xi_i$.*

Lemma 2. *If, at round k, $\mathcal{P}_i$ succeeds in performing $CP_{l,i,j} = CP(d_{l-1}, u_{i,j}, d_{l,j}, d_j)$, $l < k$, at Step 1c of Protocol 2, then $(u_{i,j}, v_{i,j})$ vetoes the face-down card $c_{k,0,j}$ at Step 5b of Protocol 1.*

Lemma 3. *A player can only veto a card she has drawn.*

Lemma 4. *The number of cards vetoed by a player is at least the number of cards drawn by the player.*

Theorem 1. *The set of cards vetoed by a player is the same as the set of cards drawn by the player.*

Theorem 2. *A vetoed card cannot be opened.*

5 Conclusions

We have presented a mental poker protocol which, to our best knowledge, is the first TTP-free proposal tolerating both intentional and accidental player dropout. Future research will explore applications of the protocol in this paper to secure multi-party computation problems other than mental poker.

References

1. A. Barnett and N. Smart, "Mental poker revisited", in *Proc. Cryptography and Coding*, LNCS 2898, pp. 370–383, December, 2003.
2. D. Chaum and T. Pedersen, "Wallet databases with observers" in *Advances in Cryptology – CRYPTO'92* , LNCS 740, pp. 89–105, 1992.
3. R.Cramer, I.Damgård and B.Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols", in *Advances in Cryptology – CRYPTO'94* , LNCS 839, pp. 174–187, 1994.
4. C. Crépeau, "A secure poker protocol that minimizes the effect of player coalitions", in *Advances in Cryptology - CRYPTO'85*, LNCS 218, pp. 73-86, 1986.

5. C. Crépeau, "A zero-knowledge poker protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face", in *Advances in Cryptology - CRYPTO'86*, LNCS 263, pp. 239-250, 1986.
6. Y. Desmedt and Y. Frankel, "Threshold cryptosystems", in *Advances in Cryptology – CRYPTO'89*, LNCS 335, pp. 307–315, 1990.
7. K. Kurosawa, Y. Katayama and W. Ogata, "Reshufflable and laziness tolerant mental card game protocol", *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*, vol. E00-A, 1997.
8. T. P. Pedersen, "A Threshold cryptosystem without a trusted party", in *Advances in Cryptology - EUROCRYPT'91*, LNCS 547, pp. 522-526, 1992.
9. C. Schindelhauer, "A toolbox for mental card games", Medizinische Universität Lübeck, 1998. `http://citeseer.nj.nec.com/schindelhauer98toolbox.html`
10. W. H. Soo, A. Samsudin and A. Goh, "Efficient mental card shuffling via optimised arbitrary-sized based permutation network", in *Information Security*, LNCS 2433, pp. 446-458, 2002.
11. M. Stadler, "Public verifiable secret sharing", in *Advances in Cryptology - EUROCRYPT'96*, LNCS 1070, pp 190-199, 1996.
12. M. Yung, "Cryptoprotocols: subscription to a public key, the secret blocking and the multi-player mental poker game", in *Advances in Cryptology- CRYPTO'84*, LNCS 196, pp 439-453, 1985.

Appendix

Proof (Lemma 1). A face-down card $c_{k,0,j}$ is formed by a set of ciphertexts $(e_{k,1,j}, \cdots, e_{k,n,j})$. When a card is *not* vetoed in Protocol 2, a re-masking factor $(u_{i,j}, v_{i,j})$ is used which still allows recovery of the cleartexts from the card ciphertexts. To allow correct decryption, the re-masking factor must satisfy $\log_g u_{i,j} = \log_y v_{i,j}$. This is exactly the property proven by $CP(g, y, u_{i,j}, v_{i,j})$. □

Proof (Lemma 2). Let us assume a card drawn in a dealing round l previous to the current round k (*i.e.* $l < k$). Let the face-up value of that card be $d_{k,j}$. $CP(d_{l-1}, u_{i,j}, d_{l,j}, d_{k,j})$ proves that:

$$\tau = \log_{d_{l-1,j}}(d_{l,j}) = \log_{u_{i,j}} d_{k,j} \tag{1}$$

From Equation 1 we have

$$u_{i,j} = (d_{k,j})^{\tau^{-1}} \tag{2}$$

Now if $(u_{i,j}, v_{i,j})$ does not actually veto $d_{k,j}$, the following holds:

$$\log_g(u_{i,j}) = \log_y(v_{i,j}) = \log_{g^\alpha}(v_{i,j}) = \frac{1}{\alpha}\log_g(v_{i,j})$$

The above is equivalent to

$$\alpha \cdot \log_g(u_{i,j}) = \log_g(v_{i,j}) \tag{3}$$

Combining Equations (2) and (3) yields

$$\log_g((d_{k,j})^{\tau^{-1}})^\alpha = \log_g v_{i,j} \tag{4}$$

If logarithms are removed, we get $v_{i,j} = ((d_{k,j})^{\tau^{-1}})^{\alpha}$. Thus, re-masking factor $(u_{i,j}, v_{i,j})$ will pass $CP(d_{l-1}, u_{i,j}, d_{l,j}, d_{k,j})$ without actually vetoing $d_{k,j}$ only if it has the form

$$(u_{i,j}, v_{i,j}) = (d_{k,j}^{\tau^{-1}}, ((d_{k,j})^{\tau^{-1}})^{\alpha}) \tag{5}$$

However, computing $v_{i,j}$ in Expression (5) without knowledge of α nor $(d_{k,j})^{\alpha}$ is as hard as the Diffie-Hellman problem. Obtaining α from the public key y is as hard as the discrete logarithm problem. Thus, passing the verification at Step 1c of Protocol 2 implies that card $d_{k,j}$ is actually vetoed. □

Proof (Lemma 3). Assume that the current dealing round is round k; let $\{d_{1,j}, \cdots, d_{l,j}, \cdots, d_{k,j}\}$ be the expressions for the j-th card at each dealing round l, where $\tau_{l,j} = \log_{d_{l-1,j}} d_{l,j}$ for $1 \leq l < k$. Assume now that $\mathcal{P}_i$ wants to construct a proof of veto for $d_{t,j}$, for some $t < k$. Then $\mathcal{P}_i$ needs to construct $(u_{i,j}, v_{i,j})$ so that she can perform $CP_{t,i,j} = CP(d_{t-1}, u_{i,j}, d_{t,j}, d_{k,j})$. This means $\log_{d_{t-1,j}} d_{t,j} = \log_{u_{i,j}} d_{k,j}$, which requires $u_{i,j} = d_{k,j}^{\tau_{t,j}^{-1}}$ so that $\mathcal{P}_i$ needs to know $\tau_{t,j} = \log_{d_{t-1,j}}(d_{t,j})$ But this logarithm is only known to $\mathcal{P}_i$ if she drew the card at round t (see Protocol 4). □

Proof (Lemma 4). Let us assume that $\mathcal{P}_i$ has extracted ξ_i cards in previous dealing rounds. At Step 1c of Protocol 2 $\mathcal{P}_i$ uses the proof by Cramer *et al.* [3] for the ξ_i re-masking factors corresponding to the ξ_i drawn cards. According to Lemma 2, this guarantees that the ξ_i drawn cards are vetoed. □

Proof (Theorem 1). If a re-masking factor $(u_{i,j}, v_{i,j})$ passes the proof that it is a vetoing factor for card j, then by Lemma 2 it vetoes card j. On the other hand, if a re-masking factor $(u_{i,j}, v_{i,j})$ passes the proof that it is a non-vetoing factor for card j, then by Lemma 1, it does not veto card j. Now, a re-masking factor $(u_{i,j}, v_{i,j})$ cannot at the same time veto and not veto card j. Thus, $(u_{i,j}, v_{i,j})$ cannot pass both the proof that it is a vetoing factor and the proof that it is a non-vetoing factor.

Let us assume that player $\mathcal{P}_i$ has drawn ξ_i cards. By Lemma 1, the number of cards not vetoed by $\mathcal{P}_i$ is at least $52 - \xi_i$. By Lemma 4, the number of cards vetoed by $\mathcal{P}_i$ is at least ξ_i. Therefore, $\mathcal{P}_i$ vetoes *exactly* ξ_i cards. Finally, by Lemma 3, a player can only veto cards she has drawn. Therefore the set of drawn cards is the same as the set of vetoed cards. □

Proof (Theorem 2). Without loss of generality we assume $n = 2$ players $\mathcal{P}_1$ and $\mathcal{P}_2$. Assume that a round k card $d_{k,j}$ is computed from $d_{k-1,j}$ by raising a round $k-1$ card $d_{k-1,j}$ to exponents m_1 and m_2, *i.e.* $d_{k,j} = d_{k-1,j}^{m_1+m_2}$. Note that m_i is secret and only known to $\mathcal{P}_i$, for $i = 1, 2$, whereas $d_{k,j}$ is public and obtained at Step 3 of Protocol 1.

At Step 5a of Protocol 1 the vetoing protocol is called to compute re-masking factors $g_{k,1,j} = (u_1, v_1)$ and $g_{k,2,j} = (u_2, v_2)$; at Step 5b these factors are applied to the encrypted card exponents $e_{k,j} = (e_{k,1,j}, e_{k,2,j})$ to veto card $d_{k,j}$. Now, (u_1, v_1) and (u_2, v_2) have been computed by the vetoing protocol (Protocol 2), so they satisfy

$$\log_g u_i \neq \log_y v_i \text{ for } i = 1, 2 \tag{6}$$

The computations for vetoing $d_{k,j}$ at Step 5b of Protocol 1 are:

$$e_{k,j} \cdot g_{k,j} = \{e_{k,1,j} \cdot g_{k,1,j}, e_{k,2,j} \cdot g_{k,2,j}\} = \{E_y(m_1) \cdot (u_1, v_1), E_y(m_2) \cdot (u_2, v_2)\}$$
$$= \{(g^{r_1} \cdot u_1, m_1 \cdot y^{r_1} \cdot v_1), (g^{r_2} \cdot u_2, m_2 \cdot y^{r_2} \cdot v_2)\} \tag{7}$$

Opening card $d_{k,j}$ means extracting the secret exponents m_1 and m_2 from the face-down card expression $e_{k,j} \cdot g_{k,j}$. From Expression (7), we have that

$$m_1 = \frac{m_1 \cdot y^{r_1} \cdot v_1}{(g^{r_1} \cdot u_1)^\alpha} \quad , \quad m_2 = \frac{m_2 \cdot y^{r_2} \cdot v_2}{(g^{r_2} \cdot u_2)^\alpha} \tag{8}$$

Some algebraic manipulation of Equations (8) leads to

$$\log_g u_1 = \log_y v_1 \quad , \quad \log_g u_2 = \log_y v_2 \tag{9}$$

Equations (9) contradict Equations (6). Thus, the card cannot be opened. □