Secure and Verifiable P2P Card Games

Chun-Chao Yeh
Department of Computer Science,
National Taiwan Ocean University, Taiwan
ccyeh@mail.ntou.edu.tw

Abstract

This paper presents the design of secure and verifiable P2P card games. An efficient scheme was proposed to achieve secret encoding, distribution, revealing, and verification in a fully distributed way. The proposed card shuffling scheme is based on lightweight operations such as permutation and modular addition. Compared with most of previous approaches basing on public key cryptosystems and their alike, the proposed scheme is simple and fast. This unique feature makes the proposed scheme more feasible for those resource limited mobile devices, such as PDAs and smartphones.

Keywords: P2P card game, secure, verifiable, TTP-free.

1 Introduction

Recent advances on Internet and wireless communication technologies enable seamless connections between internet users anytime and anywhere. In such an ubiquitous computing environment, internet users get more chances to connect themselves to others. Accordingly, we expect entertainments would be one of major activities in such a ubiquitous cyberspace. For example, people can join a networked (card) game when she is waiting for someone or for some events, with her mobile devices such as laptops, PDAs, or smartphones. Or, a group of people with their mobile devices can form an ac-hoc network and play networked games. In this paper, we consider how a networked (card) game can be proceeded faithfully without a trusted third-party (TTP). Moreover, a fast and low-computation scheme for card shuffling and dealing is the main issue to be considered, which is particularly important for those mobile devices with limited computing resources such as computing power, memory and battery.

Several difficulties arise when developing a fully distributed (card) game without a trusted third-party. How to conceal secrets between each players? How to prevent cheating? How to verify each player's claims? Another practical issues are such as low computation complexity, which is critical for resource limited platforms such as PDAs and smartphones. A secure scheme with highly complex operations would not be practical. For example, according to [12], the author found that some proposed schemes might result in eight hours to shuffle a deck, based on a study conducted by [7] in 1994. Although current processors can run much faster, say 100 times faster than the one used fourteen years ago, it still could take more or less few minutes to do the job. Apparently, it is too much as well.

Most of TTP-free card game protocols use public key cryptosytems or their alike. This paper presents a scheme to achieve fast and secret encoding, distribution, revealing, and verification in a fully distributed way. Instead of using complex cryptosystems for card dealing and shuffling, the proposed scheme utilize a secret sharing mechanism which is based on lightweight operations such as permutation and modular additive operations only. The proposed scheme can be a substrate not only for varieties of card games but also for other types of games which share similar properties to conceal/reveal secrets during the game.

The rest of the paper is organized as follows. In the following section, a quick review of related work is presented. In Section 3, we give a general description about the system model and the framework of the proposed P2P card games. In Section 4, we present the proposed algorithm. In Section 5, we prove the correctness and the security properties of the proposed algorithm. Concluding remarks are given in Section 6.

2 Related Work

The issues on how to play "electronic" card games remotely have been studied before the invention of internet. The problem is referred to as "Mental Poker" in literature, following the work conducted by Shamir, Rivest and Adleman in 1979[10]. The authors in [10] proposed a protocol to solve the problem. However, it was shown to be insecure as the proposed protocol can leak partial information of the

978-0-7695-3492-3/08 $25.00 © 2008 IEEE
DOI 10.1109/EUC.2008.165

cards[8]. Meanwhile, most of early works are focus on two-player games. Claude Crepeau[5] proposed a design framework to minimize the effect of player coalitions for multiple players. The authors proposed RSA[9] and probabilistic encryption scheme of Blum and Goldwasser[2] to implement the protocol required in the proposed framework. The same author later extend the framework to be able to conceal the players's strategy[6]. However, [12] reported that the proposed scheme seemed too complicate to be practical, based on an implementation result shown in [7].

Recently, Zhao *et al.*[15] proposed an mental poker protocol based on ElGamal cryptosystems. The proposed scheme is efficient, fast and TTP-free, compared with previous works. Unfortunately, the proposed scheme was found to be insecure under known-cleartext attacks[3]. Zhao *et al.*[14] revised the original scheme to remove the security flaw pointed out in [3]. However, the revised version introduced another security flaw pointed out in [4]. Soo *et al.*[11] proposed a scheme based on optimized arbitrary-sized Benes permutation networks to shuffle the deck. The proposed scheme uses ElGamal public key cryptosystems and takes advantage of homomorphic properties of ElGamal for re-encryption. Meanwhile, the proposed scheme can tolerate a certain number of player drop-out at the cost of multiple stages of permutation networks. Barnett *et al.* [1] proposed a cryptographic protocol, named VTMF (verifiable l-out-of-l threshold masking function). Two possible implementations (based on ElGamal encryption scheme and Paillier probabilistic encryption function respectively) of the proposed protocol were proposed.

Wierzbicki *et al.*[13] presented a P2P game, named P2P Scrabble, in which each letter X is divided into $(n+k)$ parts $\{p_{x,i}\}_{1\leq i\leq (n+k)}$, where just n players are enought to reconstruct the letter back. The remaining k parts, held by other peers, could be considered as spare. The reconstruction scheme is based on modular additive operations over n shares of secrets. A commitment function $F(x)$ is used to produce a proof of a choice a player made, which should be revealed later to prevent cheating. Compared with costly operations needed for public key cryptosystems mentioned above, the operations for revealing shared secrets with modular addition only is much much simple. However the proposed scheme relays on secret distributors to initiate the secret construction. Any player colludes with the secret distributors can reveal all the secrets.

3 Card Game Model

Assume there are p ($p > 2$) players playing a card game consisting of m ($m \geq p$) cards. Then, the card game under the proposed system framework proceeds as follows: 1. shuffling the cards, 2. playing the game, 3. verifying the game.

3.1 Card representation and shuffling

Each card can be represented by a pair($index$, $value$). Assume the card with index i, denoted as c_i, is associated with a value v_i,$0 \leq v_i \leq m-1$. The card shuffling procedure is intended to hide the secret information including both of card value of any card c_i and the binding between the value of a card c_i to its index i. For each card c_i, the value associated with the card, v_i, is encoded into n shares $\{e_{i,j} : 0 \leq j \leq n-1\}$, where $n = p$ (the number of players) if p is even number; otherwise $n = p+1$. The value v_i can be revealed only if a player gets all the share secrets $\{e_{i,j} : 0 \leq j \leq n-1\}$ of the card c_i. Since the encoding procedure is conducted by all the players, a player can reveal the secret if she either knows the final binding of the index to the value, or gets the chance to collect all the share secrets $\{e_{i,j} : 0 \leq j \leq n-1\}$ for the card c_i. How to prevent either of the two cases happening is the main concerns in this paper.

3.2 Card drawing

While detailed procedure of the card game depends on the rules of the game, in general there are two major operations involved: draw a card and show a card. To draw a card from public or from other players, is equivalent to transfer the card ownership. To show a card is equivalent to reveal the secrets to some players or to the public.

Initially each player p_k keeps one share of secrets for each card c_i. That is, each player p_k holds $\{e_{i,k} : 0 \leq i \leq m-1\}$. Before starting playing the game, if card c_i is needed to be revealed to the public, each player p_k sends $e_{i,k}$ to each others. On the other hand, if the card c_i is needed to be revealed to player p_k only, the player requests the missing parts of all $e_{i,k}$ from others.

3.3 Verifying the game

All the secrets are constructed by all players cooperatively. To prevent any player from modifying the secrets or cheating later on, a verification procedure is designed to verify each player's claims during the game. For each player, all the secrets made by the player during the card shuffling stage is organized into a predefined message format. A message digest for the message containing all the information associated with the secret of a card value is made by the player. Then, the message digest is sent to the public (that is all the players) for verification. When the game is done, all players should reveal their secret information to the public. By checking the message digest with the secret information, all players can check if anyone commits cheating.

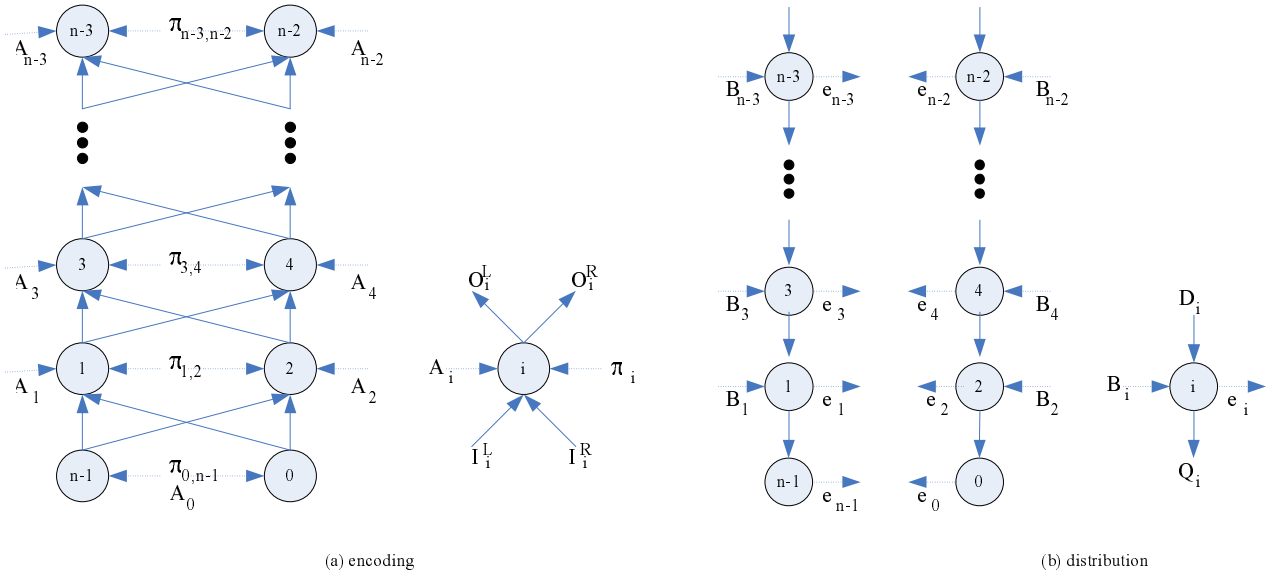

Figure 1. structure of secret encoding and distribution

notation	description
A_i	an $mxn/2$ matrix pre-determined by peer p_i, $0 < i < n-1$. $f_m(A_i) = \mathbf{0}_{mx1}$.
B_i	an mxk matrix pre-determined by peer p_i located at layer k, $0 < i < n-1$. $f_m(B_i) = \mathbf{0}_{mx1}$.
A_0	an mxn matrix pre-determined by peers p_0 and p_{n-1} cooperatively. $f_m(A_0) = \mathbf{0}_{mx1}$.
C	an mxn matrix, defined by $c_{i,j} = i$ if $j = 0$; otherwise 0.
$I_i^L(, I_i^R)$	input matrices received by peer p_i from left(,right) peers at its immediated previous layer, during secret encoding phase.
$O_i^L(, O_i^R)$	output matrices sent by peer p_i to left(,right) peers at its immediated next layer, during secret encoding.
D_i	an mxk input matrix (from layer k+1) received by peer p_i located at layer k, $0 < i < n-1$, during secret distribution phase.
Q_i	an $mx(k-1)$ output matrix (to layer k-1) generated by peer p_i located at layer k, $0 < i < n-1$, during secret distribution phase.
e_i	an $mx1$ matrix generated by peer p_i as the shared secret for p_i, $0 \leq i \leq n-1$, during secret distribution.

Table 1. related information generated/used by peer i in the proposed algorithm.

4 Proposed Schemes

4.1 Assumptions

The proposed scheme requires at least three players to construct the secret. All participants include all the game players only. No other (trusted) third-party is needed. To apply the proposed scheme to two-player games, additional player is required to construct the secret cooperatively. Meanwhile, we assume messages sent to each player are encrypted and cannot be revealed by third-party. Also, we assume players cannot collude but might commit cheating on their own. Except allowed by the game rules, a game can hardly proceed fairly when collusion happens, no matter how perfectly the game rules and the framework are designed.

4.2 Secret encoding and distribution

In the following context, we present the key design schemes on the shared secret encoding and distribution at the card shuffling stage. Figure 1 shows the structure of secret encoding and distribution. Detailed algorithm is presented in Figure 2. Related data generated/used by each peer are defined in Table 1. Meanwhile, a set of functions are used in the proposed algorithm. The row permutation function $\pi_{i,j}$ is determined by both of the two peers p_i and p_j, which reorder the rows of a given matrix. The row element permutation function R not changes the row ordering. Instead, it randomly changes the order of the elements in the same row. Each row is permutated independently. The ma-

trix partition function $P(M_{\mathtt{axb}}, c)$ is defined to partition the input matrix $M_{\mathtt{axb}}$ into two sub matrices $M'_{\mathtt{axc}}$ and $M''_{\mathtt{ax(b-c)}}$, such that $M = M' \parallel M''$. The row element summation function $f_m(M_{\mathtt{axb}})$ is defined to generate an ax1 matrix. Let $f_m(M_{\mathtt{axb}}) = V_{\mathtt{ax1}}$, then by definition, $v_{i,0} = (\sum_{0 \le k \le b-1} m_{i,k}) \bmod m$.

The proposed algorithm requires even number of peers (n) to construct the secrets. Assume the card game includes $p\,(>2)$players with player ids 0 to $p-1$, and $m\,(\ge p)$ cards with card ids 0 to $m-1$. If p is even, set $n = p$ and let player i takes the job of peer i, $0 \le i \le n-1$. On the other hand, if p is odd, set $n = p+1$. Then, the player with id 0, p_0, takes the jobs of both peer 0 and peer $n-1$ at the same time. Following the steps of the proposed scheme shown in Figure 2, each player k (except player p_0 under the case of odd number of players) holds one share of secret $e_{i,k}$ for each card c_i, $0 \le i \le m-1$. For the case p is odd, player p_0 will own two shares of secret, $\mathbf{e_0}$ and $\mathbf{e_{n-1}}$, where $\mathbf{e_k}$ denotes an mx1 matrix of $[e_{0,k}, e_{1,k}, ..., e_{m-1,k}]^t$. The player p_0 keeps one share of secret ($\mathbf{e_0}$) and makes the other one ($\mathbf{e_{n-1}}$) known to all other players. Revealing the value of card c_i is done by collecting all the shared secrets $e_{i,j}$ from all peers, and then decoding the secure with the modular additive operation: $v_i = (\sum_{0 \le j \le n-1} e_{i,j}) \bmod n$.

5 Correctness and Security

5.1 Correctness

Given n peers and m cards, the proposed algorithm(see Figure 2) generates shared secrets $\mathbf{e_k}$, an mx1 matrix of $[e_{0,k}, e_{1,k}, ..., e_{m-1,k}]^t$, for each peer k, such that all the card values can be reconstructed from the shared secrets held in each peer's hands. Consequently, for any possible card value $v, 0 \le v \le m-1$ there exists unique index i such that $v = (\sum_{0 \le k \le n-1} e_{i,k}) \bmod n$. That is,

$$f_m(\mathbf{e_0} \parallel \mathbf{e_1} \parallel ... \parallel \mathbf{e_{n-1}}) = \pi_s([0, 1, ..., m-1]^t)$$

,for some permutation function π_s. In the following context, we prove this property holds.

Lemma 1. In the proposed algorithm (Figure 2), the following properties hold for each peer i:

1. $f_m(O_i^L \parallel O_i^R) = \pi_i(f_m(I_i^L \parallel I_i^R))$ (the encoding procedure in Figure 1).
2. $f_m(\mathbf{e_i} \parallel Q_i) = f_m(D_i)$(the distribution procedure in Figure 1).

Lemma 2. In Step 3 (encoding shared secrets) of the proposed algorithm (Figure 2), the following properties hold:

1. For the two peers $(n-1, 0)$ at first layer, $f_m(O_{n-1}^L \parallel O_{n-1}^R \parallel O_0^L \parallel O_0^R) = \pi_{0,n-1}([0, 1, ..., m-1]^t)$.
2. For two peers $(i, i+1)$ at same layer $k (k > 1)$, $f_m(O_i^L \parallel O_i^R \parallel O_{i+1}^L \parallel O_{i+1}^R) = \pi_{i,i+1}(f_m(I_i^L \parallel I_i^R \parallel I_{i+1}^L \parallel I_{i+1}^R))$.

Lemma 3. In the proposed algorithm (Figure 2), the following properties hold:

1. $f_m(O_{n-3}^L \parallel O_{n-3}^R \parallel O_{n-2}^L \parallel O_{n-2}^R) = \pi_{n-3,n-2} \circ \ldots \circ \pi_{1,2} \circ \pi_{0,n-1}([0, 1, ..., m-1]^t)$.
2. $f_m(D_{n-3} \parallel D_{n-2}) = f_m(\mathbf{e_0} \parallel \mathbf{e_1} \parallel ... \parallel \mathbf{e_{n-1}})$.

Theorem 1. In the proposed algorithm (Figure 2), there exists a permutation function π_s such that $f_m(\mathbf{e_0} \parallel \mathbf{e_1} \parallel ... \parallel \mathbf{e_{n-1}}) = \pi_s([0, 1, ..., m-1]^t)$.
proof:
Since $D_{n-3} = O_{n-3}^L \parallel O_{n-3}^R$ and $D_{n-2} = O_{n-2}^L \parallel O_{n-2}^R$, from **Lemma 3** we have

$$\begin{aligned} & f_m(\mathbf{e_0} \parallel \mathbf{e_1} \parallel ... \parallel \mathbf{e_{n-1}}) \\ = \; & f_m(D_{n-3} \parallel D_{n-2}) \\ = \; & f_m(O_{n-3}^L \parallel O_{n-3}^R \parallel O_{n-2}^L \parallel O_{n-2}^R) \\ = \; & \pi_{n-3,n-2} \circ \ldots \circ \pi_{1,2} \circ \pi_{0,n-1}([0, 1, ..., m-1]^t) \end{aligned}$$

That is, there exists a permutation function $\pi_s = \pi_{n-3,n-2} \circ \ldots \circ \pi_{1,2} \circ \pi_{0,n-1}$ such that $f_m(\mathbf{e_0} \parallel \mathbf{e_1} \parallel ... \parallel \mathbf{e_{n-1}}) = \pi_s([0, 1, ..., m-1]^t)$.

5.2 Security

Since the encoding is conducted by all the players, a player can reveal the secret if she either knows the final binding of the index (that is, the result of composite permutation functions $\pi_s = \pi_{n-3,n-2} \circ \ldots \circ \pi_{1,2} \circ \pi_{0,n-1}$), or gets the chance to collect all the shares secrets $\{e_{i,j} : 0 \le j \le n-1\}$ for card c_i. Since during the secret encoding (Step 3 in the proposed algorithm) each peer holds only one of the n/2 permutation functions in the composite permutation function $\pi_s = \pi_{n-3,n-2} \circ \ldots \circ \pi_{1,2} \circ \pi_{0,n-1}$, no peer has knowledge to construct the composite permutation function and thus knows the final binding of the card index and its value. Meanwhile, during the secret distribution (Step 4 in the proposed algorithm) each peer knows only partial shared secrets $e_{i,j}$ for each card c_i, consequently no peer has knowledge to reveal the value of each card c_i without collusion with other players. As a result, under no collusion assumption, the proposed algorithm guarantees no player can reveal any card value.

Algorithm Shared_secret_encoding_distribution
Input:
System: p players, m cards
peer_i : A_i, B_i, π_i
constant matrix: C
Output: secrets $\{e_k | 0<=k<n\}$
Notations and symbols: Referred to Table 1.

Steps:
1. construct peer group
(a) determine the peer id (0, ⋯, p-1) for each player. Let peer with id k denoted as p_k.
(b) if p is odd, set n=p+1; otherwise n=p. The player with peer id 0 should take additional jobs for peer n if p is odd;
(c) The n peers form n/2 pairs (n-1, 0), (1, 2), (3, 4), ⋯ (n-3, n-2) and organize as n/2 layer of graph as shown in Figure 1.
2. Prepare private secrets
(a) Each peer k (except peer 0 and n-1) independently determines two matrices A_k and B_k such that $f_m(A_k)=f_m(B_k)=0_{mx1}$. The two peers, peer 0 and n-1, determine a common matrix A_0 such that $f_m(A_{0,})=0_{mx1}$, and set $B_0=B_{n-1}=0_{mx1}$.
(b) Each of the two peers (p_i, p_j) at same layer determines a common row permutation function $\pi_j=\pi_i=\pi_{i,j}$.
3. encode shared secrets:
(a) First, each of the two peers at first layer (layer 1) gets a new matrix $M=R(\pi_0 (C+A_0))$. Let $P(M, n/2)=M_1 \| M_2$, $P(M_1, n/4)=O_{L1} \| O_{R1}$, $P(M_2, n/4)=O_{L2} \| O_{R2}$. Send $O_{L1} \| O_{R2}$ ($O_{R1} \| O_{L2}$) to the left (right) peer in next layer (layer 2).
(b) for i= 2 to n/2 do :
each peer k at layer i>1 receives two matrices I_L and I_R from its immediate previous layer (layer i-1). And, then generates a new (mxn/2) matrix $M= R(\pi_k ([I_L \| I_R]+A_k))$. Let $P(M, n/4)=O_L \| O_R$. Send O_L (O_R) to the left (right) peer in next layer (layer i+1), if k is odd; otherwise, send O_R (O_L) to the left (right) peer in next layer (layer i+1),
4. distribute shared secrets:
(a) for peer k at top layer (layer n/2), set $D_k = O_L \| O_R$, where both O_L and O_R are generated at step 3, then generate a new matrix $M= R(D_k+B_k)$. Let $P(M, 1)=e_k \| Q_k$.
Send Q_k to the peers in the next layer (layer i-1).
(b) for i= n/2 down to 2 do:
each peer k at layer i>1 receives D_k from its immediate previous layer (layer i+1), and then generates a new matrix $M= R(D_k+B_k)$.
Let $P(M, 1)=e_k \| Q_k$. Send Q_k to the peers in next layer (layer i-1).
(c) for peers k at first layer (layer 1), take $D_k =e_k$.
5. make and announce verification message
(a) each peer k, 0≤ i< n, associates with parameters (π_k,A_k,I_L^k, I_R^k,O_L^k,O_R^k) for shared secret encoding, and parameters (B_k,D_k,e_k,Q_k) for shared secret distribution.
(b) concatenate all the parameters with the peer id and a nonce to form a message:
$s_k=k\|\pi_k\|A_k\|I_L^k\|\ I_R^k\|O_L^k\|O_R^k\|B_k\|D_k\|e_k\|Q_k\|nonce_k$.
(c) get the hash value $h_k=Hash(s_k)$ of the message s_k.
(d) send the hash value with peer id and the nonce, that is (h_k, k, $nonce_k$), to all the peers.

Figure 2. proposed algorithm for secret encoding and distribution

6 Concluding Remarks

This paper presents an efficient scheme to achieve secret encoding, distribution, revealing, and verification in a fully distributed way without a trusted third-party. Using a trusted third-party as a dealer to manage all secrets of the cards can simplify card game protocols. However, we argue that this type of setting is insecure since either the dealer can be bribed or the software served as the dealer can be attacked. Thus, the attacker gets the chances to collect all the secrets. Meanwhile, the proposed card shuffling scheme is lightweight, compared with most of previous works. Most of card shuffling schemes proposed before are based on public key cryptosystems and their alike. In this paper, instead, we proposed using modular addition and permutation to perform card shuffling, and thus reduced the computation complexity. This unique feature makes the proposed scheme more feasible for those resource limited mobile devices, such as PDAs and smartphones. Nonetheless, the proposed scheme has some weaknesses to be improved in the future. First, the proposed scheme requires at least three players involved in the game. While many card games are designed for multiple players, for those two-player games the proposed scheme requires one additional player to assist in constructing the secrets. Also, the proposed scheme has low tolerance to resist collusion. At worst case, collusion between two particular players (but not true for any two players) might reveal all share secrets. How to improve these weak points deserves more efforts to explore.

Acknowledgement

This research was supported in part by National Science Council of Taiwan under the grand NSC96-2221-E-019-009, NSC97-221-E-019-018 and by National Taiwan Ocean University under the grand NTOU-RD961-05-02-01-01.

References

[1] A. Barnett and N. P. Smart. Mental Poker Revisited. In K. G. Paterson, editor, *Cryptography and Coding, Proceedings of the 9th IMA International Conference*, volume 2898 of *Lecture Notes in Computer Science*, pages 370–383. Springer Verlag, 2003.

[2] M. Blum and S. Goldwasser. An efficient probabilistic public key encryption scheme which hides all partial information. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 289–302, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[3] J. Castellà-Roca and J. Domingo-Ferrer. On the Security of an Efficient TTP-Free Mental Poker Protocol. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '04)*, volume 2, pages 781–784. IEEE Computer Society, 2004.

[4] J. Castellà-Roca, J. Domingo-Ferrer, and F. Sebé. On the Security of a Repaired Mental Poker Protocol. In *Information Technology: New Generations, Proceedings of the Third International Conference (ITNG 2006)*, pages 664–668, 2006.

[5] C. Crépeau. A Secure Poker Protocol that Minimizes the Effect of Player Coalitions. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO '85: Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 73–86. Springer Verlag, 1985.

[6] C. Crépeau. A Zero-Knowledge Poker Protocol that Achieves Confidentiality of the Players' Strategy or How to Achieve an Electronic Poker Face. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86: Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 239–247. Springer Verlag, 1986.

[7] J. Edwards. Implementing Electronic Poker: A Practical Exercise in Zero-Knowledge Interactive Proofs. Master's thesis, Department of Computer Science, University of Kentucky, 1994.

[8] S. Goldwasser and S. Micali. Probabilistic Encryption & How To Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC '82)*, pages 365–377. ACM Press, 1982.

[9] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26(1):96–99, 1983.

[10] A. Shamir, R. L. Rivest, and L. M. Adleman. Mental Poker. Technical Report MIT-LCS-TM-125, Massachusetts Institute of Technology, 1979.

[11] W. H. Soo, A. Samsudin, and A. Goh. Efficient Mental Card Shuffling via Optimised Arbitrary-Sized Benes Permutation Network. In A. H. Chan and V. Gligor, editors, *Information Security, Proceedings of the 5th International Conference (ISC 2002)*, volume 2433 of *Lecture Notes in Computer Science*, pages 446–458. Springer Verlag, 2002.

[12] H. Stamer. Efficient Electronic Gambling: An Extended Implementation of the Toolbox for Mental Card Games. In C. Wolf, S. Lucks, and P.-W. Yau, editors, *Proceedings of the 1st Western European Workshop on Research in Cryptology (WEWoRC 2005)*, volume P-74 of *Lecture Notes in Informatics*, pages 1–12. Gesellschaft für Informatik e.V., 2005.

[13] A. Wierzbicki and T. Kucharski. "p2p scrabble. can p2p games commence?". In *Proceedings of Fourth International Conference on Peer-to-Peer Computing*, pages 100–107, Aug. 2004.

[14] W. Zhao and V. Varadharajan. Efficient TTP-Free Mental Poker Protocols. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '05)*, volume 1, pages 745–750. IEEE Computer Society, 2005.

[15] W. Zhao, V. Varadharajan, and Y. Mu. A Secure Mental Poker Protocol Over The Internet. In C. Johnson, P. Montague, and C. Steketee, editors, *ACSW Frontiers 2003, Proceedings of the Australasian Information Security Workshop*, volume 21 of *Conferences in Research and Practice in Information Technology*, pages 105–109. Australian Computer Society, 2003.