PAPER *Special Section on Cryptography and Information Security*

Reshufflable and laziness tolerant mental card game protocol

Kaoru KUROSAWA[†], *Member*, **Yutaka KATAYAMA**[††], *Nonmember*, *and* **Wakaha OGATA**[†††], *Member*

SUMMARY This paper presents a reshufflable and laziness tolerant mental card game protocol. First, our protocol can reshuffle any subset of cards. For example, some opened cards and some face down cards can be shuffled together. Next, we consider two types of honest players, currently active and currently nonactive. A player is currently nonactive if he dropped out the game or he declared "pass" and has not declared "rejoin" yet. In the proposed protocol, if more than half of the players are currently active, they can play the game. In this case, the privacy of the currently nonactive players are kept secret.

key words: mental poker, zero knowledge, multiparty protocol, probabilistic encryption, laziness tolerant

1. Introduction

Can we play card games by telephone? Many attempts have been made to answer this question. In the last paper in his series of works, Crepeau developed an exellent cryptographic (or mental) poker protocol which hides even the strategy of each player [5]. However, this protocol still does not simulate actual card games perfectly. In actual carg game, when all cards are used up, the discarded cards are shuffled and used as a new deck. Furthermore, each player can drop out in poker, for example, because the bet is too high. The dropout players do nothing in the rest of the game. In "shichi narabe", each player can "pass" up to three times. If a player declares "pass", he does nothing in that period. However, in the protocol of [5], the discarded cards cannot be reshuffled and all players must be active during the whole game. Even if just one player stops his job, other players cannot get cards from the deck.

On the other hand, a fault tolerant multiparty protocol does exist for any boolean function. If less than half of the players are malicious, the rest of the players can compute the value of the function by revealing the secret information of malicious players. In this protocol, even a temporally out of order player is considered as malicious. Therefore, the secret information even of such a nonactive player is revealed. Further, this kind of protocol is very complicated.

[†]The author is with Tokyo Institute of Technology, 2-12-1, O-okayama, Meguroku,Tokyo,152 Japan

[††]The author is with Mitsubishi Electric Corporation, 5-1-1, Ofuna, Kamakura-shi, Kanagawa, 247 Japan

[†††]The author is with Himeji Institute of Technology, 2167, Syosya, Himeji-shi, Hyogo, 671-22 Japan

We consider two types of honest players, currently active and currently nonactive. A player is currently nonactive if he dropped out the game or he declared "pass" and has not declared "rejoin" yet. The privacy of a nonactive player should be kept secret. If we apply a general purpose multiparty protocol to a mental poker, however, the privacy of a nonactive player is revealed. In other words, a player who wants to play the game at all must always be active during the game.

This paper presents a reshufflable and laziness tolerant mental card game protocol. In this protocol, if more than half of the players are currently active, they can continue the game. In this case, the privacy of nonactive players are kept secret. Nonactive players can rejoin the game later at any time they wish. Furthermore, our protocol can reshuffle any subset of cards. For example, some opened cards and some face down cards can be shuffles together. It is also as efficient as Crepeau's protocol, and it is much more efficient than general purpose multiparty protocols.

The first formal result on mental poker was given by [12]. They proved that mental poker is not possible for two players in an information theoretical sense. They also presented the first mental poker protocol in a computationally complexity theoretical sense by using commutative encryption functions. Shortly thereafter, [10] showed that at least one bit of information is leaked in the protocol of [12] (for example, the color of a card). Then, [7] introduced probabilistic encryption which keeps any partial information secret. By using this encryption, [7] showed a mental poker protocol for two players in which each player is not albe to obtain any information of his opponent's hand. Unfortunately, this scheme did not extend to more than two players. [1] and [13] showed protocols for three or more players. However, they assumed that players do not collude. [6] showed a protocol which uses a trusted third party. [4] gave a protocol for three or more players which is secure against collusion with no trusted third party. In this protocol, the k-th card in the deck is expressed by $\pi_N \cdots \pi_2 \pi_1(k)$, where π_i is a secret random permutation of player i. However, each player is asked to disclose π_i at the end of the game to verify that there was no cheating. In this case, even hands of the loose players are revealed. [5] solved this problem by introducing the "ANDOS" protocol. However, this protocol

has the ploblem mentioned earlier: all players must be active during the game and not every subset of cards cannot be reshuffled.

The probabilistic encryption scheme of [7] is based on the Quadratic residuosity problem and a plaintext m takes a value of 0 or 1. [3] generalized it to $m = 0, 1, \ldots, r-1$ by using the r^{th} residue problem, where r is a prime number. ([9] further generalized it to any r.) By combining a secret sharing scheme with the probabilistic encryption scheme of [3], [2] showed an efficient verifiable secret sharing scheme and a fault tolerant election scheme. We apply this technique to a mental card game protocol.

We suppose that there are N players. Numbers from 0 to 51 will be used to describe the cards.

2. Public key of each player

We use the 53rd residue cryptosystem of [3] because 53 is the least prime number greater than 51. Each player i chooses prime numbers p_i and q_i such that

$$53 | p_i - 1,\ 53 \not| q_i - 1 \tag{1}$$

Let $n_i = p_i q_i$. Player i also chooses y_i such that

$$y_i \neq x^{53} \bmod n_i \qquad \text{for any } x. \tag{2}$$

The probabilistic encryption scheme is described as follows.

(Secret key) p_i and q_i.

(Public key) n_i and y_i

(Plaintext) m $(0 \leqq m < 52)$

(Encryption) $C = E_i(m, x) = y_i^m x^{53} \bmod n_i$, where x is a random number.

(Decryption) For $0 \leqq j < 52$, compare $C^{(p_i-1)/53} \bmod p_i$ with $(y_i^{(p_i-1)/53})^j \bmod p_i$.

m is uniquely determined from $E_i(m, x)$ if Eq.(1) and Eq.(2) are satisfied [3]. Note that

$$E_i(m, x)^{(p_i-1)/53} = (y_i^{(p_i-1)/53})^j \bmod p_i$$

holds if and only if $m = j$ for any x. This cryptosystem has the following homomorphic property.

$$E_i(m_1, x_1) E_i(m_2, x_2) = E_i(m_1 + m_2, x_1 x_2) \tag{3}$$

Before the game starts, each player publicizes his (n_i, y_i). Each player i proves the uniqueness of the decoding to the other players by using a ZKIP (zero knowledge interactive proof system) shown in subsection 5.1.

3. Proposed scheme

3.1 Definitions and assumption

We consider a card game such that each player can drop out or he can declare "pass" and "rejoin".

Definition 3.1: A player is *dishonest* if he does not follow the protocol. A player is currently *nonactive* if he follows the protocol and dropped out or he declared pass but not yet rejoin. A player is currently *active* if he follows the protocol and is not nonactive.

A player is excluded from the game if he is detected to be dishonest.

Assumption 1: 1. All messages from each player are written on the public board.

2. Each player is a probabilistic polynomial time Turing machine.

3. For sufficiently large n_i, $\{E_i(m_1, x) \mid x \in Z_{n_i}\}$ and $\{E_i(m_2, x) \mid x \in Z_{n_i}\}$ are computationally indistinguishable for any m_1 and m_2, where E_i is given in the previous section.

4. Let $k \triangleq \lfloor (N-1)/2 \rfloor$. We assume that there are at least $k+1$ currently active players at any moment. (We restrict only the number of active players, e.g., all players can pass and rejoin in turn.)

Under this assumption, our protocol has the following properties.

R1 A player who dropped out the game does nothing in the rest of the game. A player who declared "pass" does nothing until he declares "rejoin".

R2 If all cards are used up, the discarded cards are reshuffled and used as a new deck. Any subset of cards can also be reshuffled.

3.2 Card expression

What is a card? (1) The backsides of any two cards are indistinguishable. (2) A player can identify the value of a card by opening the card.

We start by simulating this fact. Let C_t denote the card whose value is t. Let B_t be the backside of C_t. In the proposed protocol, B_t is written on the public board in such a way that (1) Even if k players collude, they can learn nothing from B_t. (2) If at least $k+1$ players are currently active, they can open a card.

B_t is expressed as follows. Let $R_t(z)$ denote a k-th order random polynomial such that $R_t(0) = t$. Then,

$$B_t = [E_1(R_t(1), x_{t,1}), \ldots, E_N(R_t(N), x_{t,N})] \tag{4}$$

where $x_{t,i}$ is a random number unknown to everynoe except player i. Each player i can learn only $R_t(i)$ from B_t. Even if k players collude, they have no information on $R_t(0) = t$ because $R_t(z)$ is a k-th order

random polynomial. Actually, (1) Form Assumption 1, the backsides of any two cards B_i and B_j are computationally indistinguishable. (2) If at least $(k+1)$ of $\{(R_t(i), x_{t,i})\}$ are opened, any player can compute $R_t(0) = t$, which is the value of B_t.

Definition 3.2: $\mathcal{B}_t \triangleq \{B_t\}$. $B_t \in_R \mathcal{B}_t$ denotes B_t which is randomly chosen from $\mathcal{B}_t$.

3.3 Shuffle

In an actual card game, 52 cards are shuffled in the following way.

(step 1) All cards are placed on the table face up. Each player checks that the card set is valid. For example, there is just one ace of spade, etc.

(step 2) All cards are turned over.

(step 3) Each player shuffles the cards in turn.

Our protocol simulates steps 1 - 3 faithfully.

Shuffle protocol

(step 1) Player 1 publicizes 52 random polynomials $\{R_i(z)\}$ such that $R_i(0) = i$ for $i = 0, 1, \ldots, 51$.

(step 2) Player 1 publicizes

$$\begin{aligned} B'_0 &\triangleq [E_1(R_0(1), 1), \ldots, E_N(R_0(N), 1)], \\ &\ldots \\ B'_{51} &\triangleq [E_1(R_{51}(1), 1), \ldots, E_N(R_{51}(N), 1)]. \end{aligned}$$

All other players check the validity of $B'_0, \ldots, B'_{51}$.

(step 3) Let

$$B'_i = (b_{i,1}, \ldots, b_{i,N}), \quad i = 0, 1, \ldots, 51.$$

Player 2 computes $\hat{B}_i \in \mathcal{B}_i$ from $B'_i \in \mathcal{B}_i$ as follows. Player 2 chooses 52 k-th order random polynomials $\{U_i(z)\}$ such that $U_i(0) = 0$, $i = 0, 1, \ldots, 51$ and random numbers $x_{i,j}$, $0 \leqq i \leqq 51, 1 \leqq j \leqq N$. Then, he computes

$$\begin{aligned} \hat{B}_i = (&b_{i,1} E_1(U_i(1), x_{i,1}), \\ &\ldots, b_{i,N} E_N(U_i(N), x_{i,N})). \end{aligned}$$

Now, player 2 chooses a random permutation π and publicizes

$$DECK_2 = (\hat{B}_{\pi(0)}, \ldots, \hat{B}_{\pi(51)}). \tag{5}$$

Finally, he proves that he has computed Eq.(5) correctly by using a ZKIP shown in Subsec. 5.2.

Player 3 - N and player 1 repeat step 3 in turn.

Randomly shuffled cards (DECK) are displayed on the public board in such a way that

$$DECK = (B_{\pi(0)}, \ldots, B_{\pi(51)}),$$

where $B_{\pi(i)} \in \mathcal{B}_{\pi(i)}$ and π is a random permutation of $\{0, 1, \ldots, 51\}$. Here, nobody knows π and the corresponding $R_i(z)$, $i = 0, 1, \ldots, 51$ of Eq.(4).

Lemma 3.1: The value of $\hat{B}_i$ is the same as that of B_i. That is, if $B'_i \in \mathcal{B}_i$, then $\hat{B}_i \in \mathcal{B}_i$.

(Proof) Let

$$\hat{R}_i(z) \triangleq R_i(z) + U_i(z).$$

Then, from Eq.(3), we have

$$\begin{aligned} b_{ij} E_j(U_i(j), x_{ij}) &= E_j(R_i(j), 1) E_j(U_i(j), x_{ij}) \\ &= E_j(R_i(j) + U_i(j), x_{ij}) \\ &= E_j(\hat{R}_i(j), x_{ij}). \end{aligned}$$

Hence,

$$\hat{B}_i = (E_1(\hat{R}_i(1), x_{i,1}), \ldots, E_N(\hat{R}_i(N), x_{i,N})).$$

Here, $\hat{R}_i(0) = R_i(0) + U_i(0) = R_i(0) = i$. □

(Remark) Any subset of cards can be shuffled in the same way. For example, we can shuffle some opened cards together with some face down cards. Furthermore, each player can similarly shuffle his own cards, but not cards of other players.

4. Elemental actions

In this section, we show how to simulate the following elemental actions.

R3 A player can get a card from the deck.

R4 A player can discard, open or give another player any card in his hand. He can also have another player choose a card form his hand.

R5 A player can shuffle his cards before R4.

Definition 4.1: A player is in the state of *getting a card* C_t if at least $k+1$ of $\{E_i(R_t(1), x_{t1}), \ldots, E_i(R_t(N), x_{tN})\}$ are written on the public board.

Definition 4.2: A player is in the state of *having a card* C_t if he knows all of $R_t(z)$ and $\{x_{ij}\}$ of Eq.(4).

First, let us outline our elemental actions. Let

$$\begin{aligned} B_{\pi(l)} = [&E_1(R_{\pi(l)}(1), x_{\pi(l),1}), \\ &\ldots, E_N(R_{\pi(l)}(N), x_{\pi(l),N})] \end{aligned}$$

R3 A player can get a card from the DECK if k other players are currently active. When player i gets the l-th card from the DECK, each currently active player j ($j \neq i$) sends $R_{\pi(l)}(j)$ to player i secretly. Then, only player i can compute the value of the card, $\pi(l) = R_{\pi(l)}(0)$, from these $\{R_{\pi(l)}(j)\}$. At this moment, he is in the state of getting the card. Next, he chooses $\hat{B} \in_R \mathcal{B}_{\pi(l)}$ by himself and writes $\hat{B}$ on the public board. Thus, he changes to the state of having the card.

R4 Let

$$HAND = (B_{i_1}, \ldots, B_{i_l})$$

be the backsides of the cards of a player's hand. When the player discards a card $B_t \in HAND$ face down, he declares, "I discard B_t". When he opens $B_t \in HAND$, he makes all of $\{(R_t(j), x_{t,j})\}$ open. When he gives $B_t \in HAND$ to player u, he sends $R_t(j)$ $(j = 1, 2, \ldots, N)$ to player u secretly. When he has another player choose a card from his hand, he has the player choose an element of his $HAND$.

R5 A player can shuffle the cards in his hand in essentially the same way as in 4.3.

Next, we show the details.

[Getting a card from the DECK]

For simplicity, suppose that player 1 gets the l-th card from the DECK, $B_{\pi(l)} \in DECK$. Let

$$B_{\pi(l)} = [E_1(R(1), x_1), E_2(R(2), x_2), \ldots, E_N(R(N), x_N)]$$

Also, suppose that players $1 \sim k+1$ are currently active.

(step 1) For $j = 2, \ldots, k+1$, player j computes $w_j \triangleq E_1(R(j), r_j)$ and writes w_j on the public board, where r_j is a random number. Let $w_1 \triangleq E_1(R(1), x_1)$. Define

$$W \triangleq (w_1, \ldots, w_{k+1})$$

At this moment, W is written on the public board.

(step 2) For $j = 2, \ldots, k+1$, player j proves the correctness of w_j by using the ZKIP of Subsec. 5.3.

(step 3) Player 1 obtains $R(j)$ by decrypting w_j for $j = 1, 2, \ldots, k+1$. Then he computes $R(0)$ from $\{R(j)\}$. This is the value of this card.

(step 4) Player 1 chooses a k-th order random polynomial $\hat{R}(z)$ such that $\hat{R}(0) = R(0)$ and writes

$$\hat{B} = [E_1(\hat{R}(1), \hat{x}_1), \ldots, E_N(\hat{R}(N), \hat{x}_N)]$$

on the public board, where $\hat{x}_i$ is a random number.

(step 5) Player 1 proves that $\hat{R}(0) = R(0)$ by using the ZKIP of Subsec. 5.4.

When step 2 is completed, player 1 is in the state of getting the card. When step 5 is completed, player 1 is in the state of having the card.

[Giving a card]

For simplicity, suppose that player 1 wants to give a card to player 2. Let the backside of the card be

$$B = [E_1(R(1), x_1), \ldots, E_N(R(N), x_N)]$$

Note that player 1 is now in the state of having the card.

(step 1) Player 1 writes

$$G = [E_2(R(1), r_1), \ldots, E_2(R(N), r_N)]$$

on the public board, where r_j is a random number.

(step 2) Player 1 proves that the same $R(z)$ is used in B and G by applying the ZKIP of Subsec. 5.3.

(step 3) Player 2 obtains $\{R(j)\}$ by decrypting G. Then, he computes $R(0)$ from $R(1)$ to $R(N)$. This is the value of this card.

(step 4) Player 2 chooses a k-th order random polynomial $\hat{R}(z)$ such that $\hat{R}(0) = R(0)$ and writes $\hat{B} = [E_1(\hat{R}(1), \hat{x}_1], \ldots, E_N(\hat{R}(N), \hat{x}_N))$ on the public board.

(step 5) Player 2 proves that $\hat{R}(0) = R(0)$ by using the ZKIP of Subsec. 5.4.

5. Related ZKIPs

In this section, we show some ZKIPs which are used in our card game protocol. In the following, the simulators of protocol 1 and 2 are constructed in a similar way to the ZKIP of QNR [8]. The simulators of protocol 3, 4 and 5 are constructed in a similar way to the ZKIP of QR [8]. All the protocols are ZKIP under no assumption.

5.1 ZKIP for the public key

Let

$$E(m, x) = y^m x^{53} \bmod n.$$

We show a ZKIP (zero knowledge interactive proof system) in which P proves to V that m is uniquely determined from $E(m, x)$.

Protocol 1

Repeat steps 1 - 4 $|n|$ times.

(step 1) V chooses m and x randomly and computes $z = E(m, x)$. He sends z to P.

(step 2) V proves that he knows m and x by protocol 2 shown below.

(step 3) P computes $\hat{m}$ from z and sends it to B.

(step 4) V checks that $\hat{m} = m$.

Suppose that z is given such that $z = y^m x^r \bmod n$. P can prove to V that she knows m and x by using the following ZKIP.

Protocol 2

Repeat steps 1 - 4 $|n|$ times.

(step 1) P chooses m' and x' randomly and computes $z' = y^{m'} x'^r \bmod n$. She sends z' to V.

(step 2) V sends a random bit e to P.

(step 3) P sends m'' and x'' to V such that $z^e z' = y^{m''} x''^r \bmod n$.

(step 4) V checks the above equation.

5.2 ZKIP for shuffle

In this subsection, we show a ZKIP which is necessary for the shuffle protocol of Subsec. 3.3. Suppose that

$$\begin{aligned} B_0 &= (b_{0,1}, \ldots, b_{0,N}), \\ &\cdots \\ B_{51} &= (b_{51,1}, \ldots, b_{51,N}) \\ DECK_2 &= (D_0, \ldots, D_{51}). \end{aligned}$$

are given. We show a ZKIP such that a prover proves that

$$(D_0, \ldots, D_{51}) = (\hat{B}_{\pi(0)}, \ldots, \hat{B}_{\pi(51)}),$$

where

$$\hat{B}_i = [b_{i,1} E_1(U_i(1), x_{i,1}), \ldots, b_{i,N} E_N(U_i(N), x_{i,N})],$$

$$U_i(0) = 0,$$

for $i = 0, 1, \ldots, 51$, where $U_i(z)$ is a k-th order random polynomial, $x_{i,j}$ is a random number, and π is a random permutation.

Protocol 3

Repeat step 1 $\sim$ step 4 n times, where n is a security parameter.

(step 1) P chooses 52 k-th order random polynomials $\{\hat{U}_i(z)\}$ such that $\hat{U}_i(0) = 0$ for $i = 0, 1, \ldots, 51$. and random numbers $\{\hat{x}_{i,j}\}$ for $0 \leqq i \leqq 51, 1 \leqq j \leqq N$. She computes

$$\begin{aligned} \tilde{B}_i = [&b_{i,1} E_1(\hat{U}_i(1), \hat{x}_{i,1}), \\ &\ldots, b_{i,N} E_N(\hat{U}_i(N), \hat{x}_{i,N})]. \end{aligned} \quad (6)$$

Then, she publicizes

$$(\hat{D}_0, \ldots, \hat{D}_{51}) = (\tilde{B}_{\tau(0)}, \ldots, \tilde{B}_{\tau(51)})$$

for a random permutation τ.

(step 2) V sends a random bit e to P.

(step 3) If $e = 0$, P opens $\{\hat{U}_i(z)\}$, $\{\hat{x}_{i,j}\}$ and τ.

If $e = 1$, P opens $\phi \triangleq \pi\tau^{-1}$, $T_i(z) \triangleq U_{\pi(i)}(z) - \hat{U}_{\pi(i)}(z)$ and $t_{i,j} \triangleq x_{\pi(i),j} / \hat{x}_{\pi(i),j}$.

(step 4) If $e = 0$, V checks that Eq.(6) holds and $\hat{U}_i(z)$ is a k-th oder polynomial. If $e = 1$, V checks that

$$d_{ij} / \hat{d}_{ij} = E_i(T_i(j), t_{ij}),$$

where

$$\begin{aligned} D_i &= (d_{i1}, \ldots, d_{iN}), \\ \hat{D}_{\phi(i)} &= (\hat{d}_{i1}, \ldots, \hat{d}_{iN}). \end{aligned}$$

and $T_i(z)$ is a k-th order polynomial.

5.3 ZKIP for equality

Let

$$\begin{aligned} u_j &\triangleq E_j(R(j), x_j) \\ w_j &= E_1(R(j), r_j). \end{aligned}$$

In this subsection, we show a ZKIP such that P proves that the plaintext of w_j is equal to that of u_j (that is, $R(j) = R(j)$).

Protocol 4

Repeat step 1 $\sim$ step 4 $|n_1|$ times.

(step 1) P chooses random numbers $\hat{x}, \hat{r}, \hat{R}$. She computes

$$\hat{u} = E_j(\hat{R}, \hat{x}), \quad (7)$$

$$\hat{w} = E_1(\hat{R}, \hat{r}). \quad (8)$$

Then, she sends $\hat{u}, \hat{w}$ to V.

(step 2) V sends a random bit e to P.

(step 3) If $e = 0$, P opens $\hat{x}, \hat{r}, \hat{R}$.

If $e = 1$, P opens $\tilde{x} \triangleq x_j \hat{x}$, $\tilde{r} \triangleq r_j \hat{r}$, and $\tilde{R} \triangleq R(j) + \hat{R}$.

(step 4) If $e = 0$, V checks that Eq.(7) and Eq.(8) hold. If $e = 1$, V checks that

$$\begin{aligned} u_j \hat{u} &= E_j(\tilde{R}, \tilde{x}), \\ w_j \hat{w} &= E_1(\tilde{R}, \tilde{r}). \end{aligned}$$

5.4 ZKIP for owning a card

Let

$$\begin{aligned} W &= (w_1, \ldots, w_{k+1}) \\ \hat{B} &= (b_1, \ldots, b_N). \end{aligned}$$

In this subsection, we show a ZKIP such that a prover proves that

$$\begin{aligned} w_j &= E_1(R(j), r_j), \quad j = 1, 2, \ldots, k+1 \\ b_j &= E_j(\hat{R}(j), \hat{x}_j), \quad j = 1, 2, \ldots, N \\ R(0) &= \hat{R}(0) \end{aligned}$$

where R and $\hat{R}$ are k-th order polynomials.

Protocol 5

Repeat steps 1 - 4 $|n_1|$ times.

(step 1) P chooses k-th order random polynomials Q and $\hat{Q}$ such that $Q(0) = \hat{Q}(0)$. P chooses random numbers s_j $(j = 1, 2, \ldots, k+1)$ and z_j $(j = 1, 2, \ldots, N)$. She computes

$$u_j \triangleq E_1(Q(j), s_j), \quad U \triangleq (u_1, \ldots, u_{k+1}), \quad (9)$$

$$c_j \triangleq E_j(\hat{Q}(j), z_j), \quad C \triangleq (c_1, \ldots, c_N). \quad (10)$$

Then, she sends U and C to V.

(step 2) V sends a random bit e to P.

(step 3) If $e = 0$, P opens $Q, \hat{Q}, \{s_j\}(j = 1, \ldots, k+1)$ and $\{z_j\}(j = 1, \ldots, N)$. If $e = 1$, P opens R',$\hat{R}'$, $\tilde{r}_j$ $(j = 1, \ldots, k+1)$ and $\tilde{x}_j$ $(j = 1, \ldots, N)$, such that

$$\tilde{r}_j \triangleq r_j s_j, \quad R' \triangleq R + Q,$$
$$\tilde{x}_j \triangleq \hat{x}_j z_j, \quad \hat{R}' \triangleq \hat{R} + \hat{Q}.$$

(step 4) If $e = 0$, V checks that Eq.(9),(10) and $Q(0) = \hat{Q}(0)$ hold. If $e = 1$, V checks that

$$u_j w_j = E_1(R'(j), \tilde{r}_j),\ c_j b_j = E_j(\hat{R}'(j), \tilde{x}_j),$$
$$\hat{R}'(0) = R'(0).$$

6. Security of the total protocol

6.1 Adversary

We consider the privacy and the correctness of the total protocol. We consider a probabilistic polynomial time dynamic adversary P who controls dishonest players. P satisfies assumption 1. P can choose a player who she will corrupt based on her knowledge so far.

6.2 Corrrectness

We define correctness as follows.
(1) There exists just one card for each number at any moment.
(2) Each card is opened correctly.
Now, it is easy to see that the soundness of each ZKIP of the previous section holds. Then from the soundness of each ZKIP, (1) holds with high probability. If (1) holds, (2) is clear from the uniqueness of the decoding.

6.3 Privacy

We will show that for any adversary A, there exists a simulator M whose output is indistinguishable from A's view of the whole game under assumption 1. A's view means everything that the dishonest players see. Therefore, we can say that the adversary learns nothing, so the privacy of honest players is kept secret.

M simulates the real protocol as faithfully as he could. Protocol 1 through 5 of section 5 are all ZKIPs. However, for example, the simulator cannot simulate step 1 of "getting a card" because he doesn't know $R(j)$ for honest j. In this case, the simulator uses a random number as $R(j)$ and computes w'_j. From assumption 1, the output is indistinguishable. Next, M must simulate the ZKIP of step 2. In this case, the input to the simulator is not legal w_j, (but w'_j which is computationally indistinguishable from w_j). We show a simulator M for w'_j of Protocol 4 below. Suppose that player j is honest. Let

$$u_j = E_j(R(j), x_j)$$

be his component of the l th card in the DECK. In the real protocol of getting a card, he opens

$$w_j = E_1(R(j), r_j)$$

at step 1 and proves its correctness at step 2. Now the simulator M doesn't know $R(j)$. So, M opens

$$w'_j = E_1(t, r_j) \quad (11)$$

at step 1, where t is a random number. Next, M repeats the following step 1 - 4 $|n_1|$ times to simulate Protocol 4 on input (u_j, w'_j) and the public keys of E_1 and E_j.

(step 1) M chooses random bit e'.

(step 2) If $e' = 0$, M follows step 1 of protocol 5. If $e' = 1$, M shooses random numbers $\tilde{x}, \tilde{r}, \tilde{R}$. It computes

$$\hat{u} = E_j(\tilde{R}, \tilde{x})/u_j,\ \hat{w} = E_1(\tilde{R}, \tilde{r})/w'_j.$$

It sends $\hat{u}, \hat{w}$ to V.

(step 3) M recieves e from V. If $e = e'$, M goes to next step, otherwise resets V and goes to step 1.

(step 4) M outputs

$$\begin{cases} (\hat{u}, \hat{w}, 0, \hat{x}, \hat{r}, \hat{R}) & \text{if } e = 0, \\ (\hat{u}, \hat{w}, 1, \tilde{x}, \tilde{r}, \tilde{R}) & \text{if } e = 1. \end{cases}$$

Now we will prove that assumption 1 doesn't hold if (u_j, w_j, Z) and (u_j, w'_j, Z') are distinguishable, where Z is the view of the adversary and Z' is the output of the simulator M. Suppose that there exists a probabilistic polynomial time distinguisher D_1 which distinguishes (u_j, w_j, Z) and (u_j, w'_j, Z') with high probability. From

D_1, we can show that there exist some m_1 and m_2 such that there exists a probabilistic polynomial time distinguisher D_0 which distinguishes (u, w, Z) and (u, w', Z') with high probability, where

$$u = E_j(m_1, x), w = E_1(m_1, r), w' = E_1(m_2, r)$$

where Z is the view of the adversary for (u, w) and Z' is the output of the simulator M for (u, w').

Now let us construct our final distinguisher D_2 which distinguishes $E_1(m_1, x)$ and $E_1(m_2, x)$. The input to D_2 is the public key of E_1 and a ciphertext α, where $\alpha = E_1(m_1, x)$ or $E_1(m_2, x)$.

(step 1) D_2 chooses a public key of E_j and computes $u = E_j(m_1, x)$, where x is a random number.

(step 2) D_2 runs M with input (u, α) and the public keys of E_1 and E_j.

(step 3) Let the output of M be Z_0. D_2 runs D_0 with input (u, α, Z_0) and the public keys of E_1 and E_j.

(step 4) D_2 outpus the output of D_0.

It is easy to see the following lemma.

Lemma 6.1: In eq.(11), if $t = R(j)$, then (u_j, w_j, Z) and (u_j, w_j, Z') have the same probability distribution, where Z is the view of the verifier and Z' is the output of the simulator M.

Then from the property of D_0, D_2 can distinguish $E_1(m_1, x)$ and $E_1(m_2, x)$ with high probability. This is against assumption 1.

We can show such a simulator for each ZKIP similarly.

References

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

[2] J. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret", Proc. CRYPTO'86, pp.251–260, 1987

[3] J. Cohen and M. Fisher, "A robust and verifiable cryptographically secure election scheme", Proc. 26th FOCS, pp.372–382

[4] C. Crepeau, "A secure poker protocol that minimizes the effect of player coalitions", Proc. CRYPTO'85, pp.73–86, 1985

[5] C. Crepeau, "A zero knowledge poker protocol that achieves confidentiality of players' strategy, or how to achieve an electronic poker face", Proc. CRYPTO'86, pp.239–247, 1986

[6] S. Fortune, and M. Merrit, "Poker protocols", Proc. CRYPTO'84, pp.454–464, 1985

[7] S. Goldwasser, and S. Micali, "Probabilistic encryption and how to play mental poker, keeping secret all partial information", 14th STOC, pp.365–377, 1982

[8] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems", SIAM J.Computing, vol.18, No.1, pp.186–208, 1989

[9] K. Kurosawa, Y. Katayama, W. Ogata, and S. Tsujii, "General public key cryptosystems and Mental poker protocols", Proc. EUROCRYPT'90, pp.374–388, 1990

[10] R. Lipton, "How to cheat at mental poker", Proc. of the AMS Short Course in Cryptography,1986

[11] A. Shamir, "How to share a secret", Commun. ACM, pp.612–613, 1979

[12] A. Shamir, R. Rivest, and L. Adleman, "Mental poker", in Mathematical Gardner, D.E.Klarner, ed., Wadsworth International, pp.37–43, 1981

[13] M. Yung, "Subscription to a public key, The Secret Blocking and the Multi-Player Mental Poker Game", Proc. CRYPTO'84, pp.439–453, 1984

Kaoru Kurosawa received the B.E. and Dr. Eng. degrees in electrical engineering in 1976 and 1981, respectively, from Tokyo Institute of Technology. He is currently an Associate Professor in the Department of Electrical and Electronic Engineering at Tokyo Institute of Technology. His research interests are information security and digital signal processing.

Yutaka Katayama was born in Niigata, Japan, in 1965. He received the B.E. degrees in Electrical and Electronic Engineering in 1989 from Tokyo Institute of Technology. He currently works at Mitsubishi Electric Co. and Dream Train Internet Inc. He has been engaged in development of computer software and interested in information security.

Wakaha Ogata received B.S. and Dr.Eng. degrees in 1989 and 1994, respectively, from Tokyo Institute of Technology. She is currently a research associate in the Department of Computer Engineering at Himeji Institute of Technology, and involved researches on information security.