On the Security of a Repaired Mental Poker Protocol

Jordi Castellà-Roca, Josep Domingo-Ferrer and Francesc Sebé
Rovira i Virgili University of Tarragona
Dept. of Computer Engineering and Maths
Av. Paisos Catalans, 26, E-43007 Tarragona, Catalonia

Abstract

In 2003, Zhao, Varadharajan and Mu proposed a mental poker protocol whose security was shown to be flawed in 2004: any player (or any outsider knowing the deck coding) is able to decrypt encrypted cards without knowing the encryption key. In 2005, the first two authors published a repaired version of this TTP-free mental poker protocol. We show here that this second version is also flawed: the first player can find all cleartexts of the final encrypted shuffled deck of cards. Both protocols are similar to Shamir-Rivest-Adleman's mental poker, but they replace an exponential commutative cipher with an ElGamal-like commutative cipher. We conclude that changing the underlying commutative cipher is the reason of their weakness.

Keywords: *Security protocols, Mental poker, Cryptanalysis.*

1. Introduction

Mental poker is a practical and complex case of secure multiparty computation. In fact, mental poker protocols are relevant even beyond the e-gaming world, because a lot of cryptographic constructions devised for mental poker can be used in other secure multiparty computation applications, like secure e-voting.

A generally accepted distinction in mental poker protocols is between those based on a trusted third-party (TTP) and those which are TTP-free. TTP-based protocols tend to be more efficient from the computational point of view and the presence of the TTP usually gives an impression of fairness to players [3]. However, already in 1985 Crépeau [4] argued that good mental poker protocols should be TTP-free. The reason is that it is unrealistic to rely on a TTP, because any human can be bribed and no machinery is entirely safe because no fully tamper-proof device has yet been produced. In the last two decades, a substantial number of contributions have been made to the literature on TTP-free mental poker protocols.

In [9], Zhao, Vadaharajan and Mu presented a TTP-free mental poker protocol which accepts more than two players. In [2], we described an attack against this protocol. In [8], Zhao and Vadaharajan have presented a repaired version of the initial protocol. We will show in this paper that the repaired version is flawed too. Both [9] and [8] are similar to Shamir-Rivest-Adleman's protocol [6], but they replace an exponential commutative cipher with an ElGamal-like commutative cipher. So this paper illustrates how changing the underlying commutative cipher can turn a safe protocol into an unsafe one.

Section 2 recalls the initial protocol in [9] and the attack in [2]. Section 3 recalls the repaired protocol in [2]. Section 4 describes our attack to the repaired protocol. Section 5 is a conclusion.

2. The Zhao-Vadaharajan-Mu 2003 protocol and its security problems

The protocol [9] is similar to the first TTP-free mental poker protocol [6] but uses the ElGamal cryptosystem [5] instead. All players use the same large prime number p so that ElGamal becomes a commutative cryptosystem. To show how it works, we can put an example in which x is first encrypted by player $\mathcal{P}_1$ and later by player $\mathcal{P}_2$; next, the cryptogram is decrypted by $\mathcal{P}_1$ and later by $\mathcal{P}_2$, who obtains x again.

1. $\mathcal{P}_1$ randomly chooses a factor r_1 and encrypts x with her private key K_1

$$E_{K_1}(x) = (y_{1,1}, y_{1,2}) \begin{cases} y_{1,1} = \alpha_1^{r_1} \bmod p \\ y_{1,2} = x\beta_1^{r_1} \bmod p \end{cases}$$

2. $\mathcal{P}_2$ in a similar way chooses at random a factor r_2

Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)
0-7695-2497-4/06 $20.00 © 2006 IEEE

and encrypts $y_{1,2}$ with her private key K_2:

$$E_{K_2}(y_{1,2}) = (y_{2,1}, y_{2,1,2}) \begin{cases} y_{2,1} = \alpha_2^{r_2} \bmod p \\ y_{2,1,2} = x\beta_1^{r_1}\beta_2^{r_2} \bmod p \end{cases}$$

3. $\mathcal{P}_1$ decrypts $y_{2,1,2}$:

$$\begin{aligned} D_{K_1}(y_{2,1,2}) &= y_{2,1,2}(y_{1,1}^{k_1})^{-1} \bmod p \\ &= x\beta_1^{r_1}\beta_2^{r_2}(\beta_1^{r_1})^{-1} \bmod p \\ &= y_{2,2} \end{aligned}$$

4. Later $\mathcal{P}_2$ decrypts $y_{2,2}$ and obtains x:

$$D_{K_2}(y_{2,2}) = y_{2,2}(y_{2,1}^{k_2})^{-1} \bmod p = x$$

Players choose a set of 52 values to represent a deck of cards. Every card is encrypted by every player in turn, and a deck of encrypted cards is obtained. When a player wants a card, the rest of players decrypt that card. The player who requested the card receives it encrypted with her key, and she secretly decrypts the card and obtains it in the clear. Next, we describe the various protocols of proposal [9], assuming there are n players, $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$.

Protocol 1 (Initialization)

1. *Players choose a large prime number p and all subsequent operations are done over Z_p;*
2. *Each player $\mathcal{P}_i$ computes her secret key pair K_i, where $K_i = \{(p, \alpha_i, k_i, \beta_i) : \beta_i \equiv \alpha_i^{k_i} \bmod p\}$;*
3. *The deck of cards is represented by 52 values, $D = \{x_1, \ldots, x_{52}\}$, which are agreed by all players.*

In Protocol 2 every player encrypts the chosen values x_i with her public key and obtains a set of cryptograms, which are sent to other players. Next, every player encrypts the sets received from other players with her public key. The players finally obtain as many sets with the same cryptograms as there are players. If one or more players are not fair, then the sets are different.

Protocol 2 (Card shuffling (D))

1. *For each $\mathcal{P}_i$ in $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ do;*
   (a) *Use Procedure 1 with D, α_i and β_i and obtain $E_{0,i}$ and r_i;*
   (b) *Publish $E_{0,i}$;*
2. *All players form the set $E_0 = \{E_{0,1}, \ldots, E_{0,n}\}$;*
3. *For each $\mathcal{P}_i$ in $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ do;*
   (a) *Receive the set $E_{i-1} = \{E_{i-1,1}, \ldots, E_{i-1,n}\}$ from $\mathcal{P}_{i-1}$;*
   (b) *Compute $\beta_i^{r_i}$ and $\alpha_i^{r_i}$;*
   (c) *For each $E_{i-1,j} = \{e_{i-1,j,1}, \ldots, e_{i-1,j,52}\}$ in $\{E_{i-1,1}, \ldots, E_{i-1,n}\} - \{E_{i-1,i}\}$ do:*
      i. *For each $e_{i-1,j,k} = (y_{i-1,j,k,1}, y_{i-1,j,k,2})$ in $E_{i-1,j}$ do:*
         A. *Compute $y_{i,j,k,2} = y_{i-1,j,k,2}\beta_i^{r_i}$;*
         B. *Compute $y_{i,j,k,1} = y_{i-1,j,k,1} \cup \{\alpha_i^{r_i}\}$;*
         C. *Compute $e'_{i,j,k} = (y_{i,j,k,1}, y_{i,j,k,2})$;*
      ii. *Compute the set $E'_{i,j} = \{e'_{i,j,1}, \ldots, e'_{i,j,52}\}$;*
      iii. *Choose a permutation π_i of 52 values;*
      iv. *Permute the order of cryptograms in $E'_{i,j}$ to obtain $E_{i,j} = \{e_{i,j,1}, \ldots, e_{i,j,52}\}$, where $e_{i,j,\pi_i(k)} = e'_{i,j,k}$;*
   (d) *Compute the set $E_i = \{E_{i,1}, \ldots, E_{i,n}\}$;*
   (e) *Publish E_i;*
4. *All players verify that $E_{n,i} \equiv E_{n,j}$ $\forall i, j \in \{1, \ldots, n\}$.*

Each player uses Procedure 1 in order to encrypt the deck of cards D. The procedure returns and encrypted and shuffled deck of cards E' and the value r that has been used to encrypt the deck.

Procedure 1 (D, α, β)

1. *Choose one value r, where $1 < r < p - 1$;*
2. *For each $x_i \in D$ encrypt x_i using r to get $e_i = (y_{i,1}, y_{i,2}) = (\alpha^r, x_i\beta^r)$; let the resulting deck of encrypted cards be $E = \{e_1, \ldots, e_{52}\}$;*
3. *Choose a permutation π of 52 values;*
4. *Permute the order of cryptograms in E to obtain $E' = \{e'_1, \ldots, e'_{52}\}$, where $e'_{\pi(i)} = e_i$;*
5. *Return E' and r.*

When a player wants a card she chooses a cryptogram in $E_{n,i}$, where $i \in \{1, \ldots, n\}$, and marks that cryptogram as chosen. Players can only take a cryptogram that is not marked. The rest of players in turn decrypt the cryptogram chosen by the player. The last player sends the cryptogram to the player that requested the card. At this point, the card value is protected only by the key of the player drawing the card.

Let us assume that $\mathcal{P}_k$ wants a card and uses Protocol 3.

Protocol 3 (Card drawing)

1. *$\mathcal{P}_k$ chooses a cryptogram $e_{n,i,l} \in E_{n,i}$ and marks it as chosen;*
2. *$\mathcal{P}_k$ sends $c_0 = e_{n,i,l}$ to the rest of players;*
3. *For each player $\mathcal{P}_j \in \{\mathcal{P}_1, \ldots, \mathcal{P}_{k-1}, \mathcal{P}_{k+1}, \ldots, \mathcal{P}_n\}$ do:*
   (a) *Receive $c_{j-1} = (y_{j-1,1}, y_{j-1,2})$;*

(b) *Compute* $c_j = (y_{j,1}, y_{j,2})$ *where* $y_{j,1} = y_{j-1,1} = \{y_{j-1,1,1}, \ldots, y_{j-1,1,n}\}$ *and* $y_{j,2} = y_{j-1,2} \cdot ((y_{j,1,j})^{k_j})^{-1} = y_{j-1,2} \cdot (((\alpha_j)^{r_j})^{k_j})^{-1}$;

(c) *Send* c_j *to the next player;*

4. $\mathcal{P}_k$ *receives* $c_n = (y_{n,1}, y_{n,2})$;
5. $\mathcal{P}_k$ *decrypts* c_n *and obtains her card* $d = y_{n,2} \cdot ((y_{n,1,k})^{k_k})^{-1}$

When the game is over, players reveal their secret random numbers $\{r_1, \ldots, r_n\}$. Every player can check whether the rest of players have been honest. This game verification reveals the strategy of players. The authors suggest to use a TTP, called Dealer, if the strategy of players is to be kept confidential. At the end of the game the Dealer receives the secret random numbers $\{r_1, \ldots, r_n\}$ and checks the fairness of the game.

2.1. Security problems

Unfortunately, the use by [9] of an ElGamal-like commutative cryptosystem introduces a basic weakness. In [2], we showed that, with little computation, a player can decrypt the encrypted cryptograms and find the cleartext cards x_i, for $i = 1$ to 52.

The attack [2] is a known-cleartext one. The cleartext card values $D = \{x_1, \ldots, x_{52}\}$ are assumed to be public or at least known to all players. However, when choosing an encrypted card or cryptogram, the player does not see the cleartext card x_i in it because this value is multiplied by a *hiding factor*, which will be denoted by f_h.

In Step 4 of Protocol 2 (shuffling protocol) it is required that all sets $E_{n,i}$ have the same elements. Therefore, in order to simplify the notation we will denote $E = E_{n,i} \forall i \in \{1, \ldots, 52\}$. This set has the following elements $E = \{e_1, \ldots, e_{52}\}$, where $e_i = (y_{i,1}, y_{i,2})$. In Step 3(c)iB of Protocol 2 it can be seen that $\alpha_i^{r_i}$ is appended to $y_{i,1}$ (denoted as $y_{i,j,k,1}$ in that Step). This operation is done $\forall i \in \{1, \ldots, 52\}$. Then we can assert that $y_{i,1} = \{\alpha_1^{r_1}, \ldots, \alpha_{52}^{r_{52}}\}$ $\forall i \in \{1, \ldots, 52\}$. Moreover, in Step 3(c)iA of Protocol 2 we see that all elements are multiplied by the same factor $\beta_i^{r_i}$. In other words, all values x_i are hidden with the same hiding factor

$$f_h = \beta_1^{r_1} \cdots \beta_n^{r_n}$$

If an attacker multiplies any encrypted card $y_{i,2}$ by f_h^{-1}, she will be able to find its cleartext value x_i without knowing the encryption key:

$$y_{i,2} \cdot f_h^{-1} = x_i(\beta_1^{r_1} \cdots \beta_n^{r_n})(\beta_1^{r_1} \cdots \beta_n^{r_n})^{-1} = x_i$$

Now it remains to show how an attacker can find and invert f_h. This computation can be done as follows:

Procedure 2 (D)

1. *Let* x_i *and* x_j $(i \neq j)$ *be two card values chosen during the initialization step;*
2. *The attacker computes the multiplicative inverses* x_i^{-1} *and* x_j^{-1} *modulo* p;
3. *The attacker multiplies each encrypted card* $y_{k,2}$, *for* $k = \{1, \cdots, 52\}$ *by* x_i^{-1} *and a first set of values* D_1 *is obtained. One of the values in this set is* f_h*:*

$$y_{i,2} \cdot x_i^{-1} = x_i \beta_1^{r_1} \cdots \beta_n^{r_n} x_i^{-1}$$
$$= \beta_1^{r_1} \cdots \beta_n^{r_n} = f_h$$

4. *The procedure of the previous step is repeated by the attacker with* x_j^{-1} *and a second set of values* D_2 *is obtained. Again, one of the values in* D_2 *is* f_h*:*

$$y_{j,2} \cdot x_j^{-1} = x_j \beta_1^{r_1} \cdots \beta_n^{r_n} x_{j,}^{-1}$$
$$= \beta_1^{r_1} \cdots \beta_n^{r_n} = f_h$$

5. *With overwhelming probability,* $D_1 \cap D_2$ *contains a single element and this element is* f_h.
6. *Once* f_h *is known, computing its multiplicative inverse* f_h^{-1} *modulo* p *is trivial.*

The cost of this attack is 52 products for Step 3 plus 52 products for Step 4; that is, 104 products are enough to decrypt the deck of cards.

3. The repaired Zhao-Vadaharajan 2005 protocol

In [8], a repaired version of [9] was presented by the first two authors of the initial protocol. We briefly review the repaired protocol in this section.

Assume there are n ordered players. The cleartext card values $D = \{x_1, \ldots, x_{52}\}$ are assumed to be public or at least known to all players. The deck of cards is shuffled for all players. Player $\mathcal{P}_1$ encrypts and permutes the values x_i in D for the first time and sends the cryptograms to $\mathcal{P}_2$. Player $\mathcal{P}_2$ re-encrypts the cryptograms. She permutes the re-encrypted cryptograms and sends them to $\mathcal{P}_3$. This process is repeated up to $\mathcal{P}_n$.

The encryption and decryption procedures used in [8] are the same of [9]. Without loss of generality, we can assume that Protocol 4 is run by $\mathcal{P}_1$ and $\mathcal{P}_2$, who have each an ElGamal-like key pair: $K_1 = \{p, \alpha_1, k_1, \beta_1 = \alpha_1^{k_1}\}$ for $\mathcal{P}_1$ and $K_2 = \{p, \alpha_2, k_2, \beta_2 = \alpha_2^{k_2}\}$ for $\mathcal{P}_2$.

Let $E_0 = \{e_{0,1}, \ldots, e_{0,52}\}$, where $e_{0,i} = \{y_{0,i,1} = \emptyset, y_{0,i,2} = x_i\}$

Protocol 4 (Card shuffling)

1. *$\mathcal{P}_1$ uses Procedure 3 with E_0, α_1 and β_1. She obtains E_1 and π_1;*
2. *$\mathcal{P}_1$ sends E_1 to $\mathcal{P}_2$;*
3. *$\mathcal{P}_2$ uses Procedure 3 with E_1, α_2 and β_2. She obtains E_2 and π_2*
4. *$\mathcal{P}_2$ publishes E_2, which is the shuffled deck of cards.*

Procedure 3 encrypts and permutes the elements in a deck E.

Procedure 3 ($E = \{e_1, \ldots, e_{52}\}$,$\alpha$,$\beta$)

1. *Choose a set of secret random numbers $R = \{r_1, \ldots, r_{52}\}$;*
2. *Choose a random permutation π of 52 values;*
3. *Let $e_i = (y_{i,1}, y_{i,2})$; for $i = 1$ to 52 do:*
   (a) *compute $y'_{i,2} = y_{i,2}\beta^{r_i} \bmod p$;*
   (b) *compute $y'_{i,1} = y_{i,1} \cup \alpha^{r_i} \bmod p$;*
   (c) *form $e'_i = (y'_{i,1}, y'_{i,2})$;*
4. *form $E' = \{e'_{\pi(1)}, \ldots, e'_{\pi(52)}\}$;*
5. *return E' and π.*

4. An attack to the repaired protocol

We will show that, even if the repaired protocol does not use the same hiding factor for all cards, the use of an ElGamal-like commutative cryptosystem still allows the first player to find all cleartexts of the final encrypted shuffled deck of cards.

The idea of the attack is that the structure of Protocol 4 allows player $\mathcal{P}_1$ to know the cleartext of cryptogram in successive iterations, so that $\mathcal{P}_1$ can track the cleartext of each cryptogram in the final shuffled deck of cards.

We examine the sets E_1 and E_2 in Protocol 3:

- We have $E_1 = \{e_{1,1}, \ldots, e_{1,52}\}$, where $e_{1,i} = \{y_{1,i,1}, y_{1,i,2}\}$ and $y_{1,i,1} = \{\alpha_1^{r_{1,i}}\}$.
- On the other hand, $E_2 = \{e_{2,1}, \ldots, e_{2,52}\}$, where $e_{2,j} = \{y_{2,j,1}, y_{2,j,2}\}$ and $y_{2,j,1} = \{\alpha_1^{r_{1,i}}, \alpha_2^{r_{2,j}}\}$.

$\mathcal{P}_1$ does not need to decrypt $e_{2,j}$ to know that $e_{2,j}$ is the encryption of a cleartext x_s. The reason is that $\mathcal{P}_1$ knows the permutation π_1. She can link $y_{1,i,1}$ to the cleartext x_s. She also can see that $e_{2,j}$ contains the element $\alpha_1^{r_{1,i}}$. Since this element also belongs to $y_{1,i,1}$, $\mathcal{P}_1$ can conclude that $e_{2,j}$ is the encryption of x_s.

In this way, player $\mathcal{P}_1$ can find the cleartexts of all cryptograms in E_2. The second element of $e_{2,j}$ reveals the value of the card. This same flaw occurs if there are n players rather than two: the first player can always determine the cleartexts of cards after these have been encrypted and shuffled by all players.

This attack could have been anticipated with an accurate security analysis. A possible solution is to use ElGamal re-masking instead of the proposed encryption and decryption procedures. This solution is used by other authors, such as [7] and [1]. However, if [8] used ElGamal re-masking, its novelty over the aforementioned two proposals would be very thin.

5. Conclusions

Replacing the exponential commutative cryptosystem used in (Shamir-Rivest-Adleman's [6]) with an ElGamal-like commutative cryptosystem in the way of [9, 8] degrades the security of the TTP-free mental poker protocol.

The weakness of [9] is more obvious because the same hiding factor is used for all cards, which allows any player (or any outsider knowing the deck coding) to decrypt encrypted cards without knowing the encryption key.

In [8], the hiding factor is not the same for all cards. Nevertheless, the use of an ElGamal-like commutative cryptosystem still allows the first player to find all cleartexts of the final encrypted shuffled deck of cards.

Acknowledgments

The authors are partly supported by the Catalan government under grant 2005 SGR 00446, and by the Spanish Ministry of Science and Education through project SEG2004-04352-C04-01 "PROPRIETAS".

References

[1] A. Barnett and N. Smart, "Mental poker revisited", in *Proc. Cryptography and Coding*, LNCS 2898, pp. 370–383, December, 2003.

[2] Some of these authors, "On the security of an efficient TTP-Free mental poker protocol", an IEEE conference, 2004.

[3] J. S. Chou and Y. S. Yeh, "Mental poker game based on a bit commitment scheme through network", *Computer Networks*, vol. 38, pp. 247-255, 2002.

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

[6] A. Shamir, R. Rivest and L. Adleman, "Mental poker", *Mathematical Gardner*, pp. 37-43, 1981.

[7] 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.

[8] W. Zhao and V. Varadharajan, "Efficient TTP-free mental poker protocols", in H. Selvaraj and P. K. Srimani, eds., *Proceedings of ITCC'2005*, Los Alamitos CA: IEEE Computing Society, vol. 1, pp. 745-750, April 2005.

[9] W. Zhao, V. Vadaharajan and Y. Mu, "A secure mental poker protocol over the Internet", in C. Johnson, P. Montague and C. Steketee, eds., *Australasian Information Security Workshop*, vol. 21 of *Conferences in Research and Practice in Information Technology*, pp. 105-109, Adelaide, Australia: Australian Computing Society, 2003.