On the Security of an Efficient TTP-Free Mental Poker Protocol

Jordi Castellà-Roca and Josep Domingo-Ferrer
Universitat Rovira i Virgili
Dept. of Computer Engineering and Maths
Av. Països Catalans 26
E-43007 Tarragona, Catalonia, Spain
e-mail {jcaste,jdomingo}@etse.urv.es

Abstract

Using commutative cryptosystems is a way to obtain efficient mental poker protocols which do not require using a Trusted Third Party (TTP). However, the security of such protocols depends on the particular cryptosystem used. We show that a TTP-free mental poker protocol using an ElGamal-like commutative cryptosystem is insecure.

Keywords: *Mental poker, E-gambling, Cryptographic protocols without Trusted Third Party (TTP).*

1. Introduction

The number of activities that can be carried out through a computer network is increasing day by day. One of such activities is e-gaming and, more specifically, casino-type e-gaming, *i.e.* e-gambling. Poker is one of the best known gambling games. Remote poker playing is nonetheless challenging from the data security viewpoint, in that it is not trivial to achieve and guarantee fairness. *Mental poker* is a name which labels all scientific work directed to achieving secure e-poker. Mental poker proposals in the literature can be split into two groups depending on whether they use a trusted third party (TTP) or not.

Using a TTP simplifies the mental poker protocols and significantly reduces their execution time. However, the use of a TTP also raises the concern about who is in control of the TTP. Mental poker proposals relying on a TTP include [Chou02, Hall97, Harn00].

Mental poker proposals without a TTP normally result in more complex protocols, because fairness must be achieved in a distributed fashion. As a result, TTP-free protocols tend to be slower and/or sometimes they are limited to two players. Proposals in this group include [Crép86, Crép87, Gold82, Kuro97]. Some recent attempts at designing TTP-free protocols whose efficiency is more similar to protocols using TTP can be found in [Zhao03, Cast03]. The protocol [Zhao03] is based on the older protocol [Sham81], while [Cast03] uses random permutations and homomorphic encryption.

1.1. Our contribution

We will show here that, even though the [Zhao03] is based on [Sham81], the use by the former of a different cryptosystem introduces a security flaw. The flaw is that a player can decrypt the reversed (encrypted) cards of other players.

Section 2 briefly recalls the protocol [Sham81] and [Zhao03], which are quite similar. Section 3 describes the proposed attack. Section 4 is a conclusion.

2. Two TTP-free mental poker protocols based on commutative encryption

Protocols [Sham81] and [Zhao03] use both commutative encryption to eliminate the need for a TTP. The difference between both proposals essentially is the cryptosystem they use. We next outline the operation of both protocols.

2.1. Shamir-Rivest-Adleman mental poker

The proposal [Sham81] is limited to two players A and B and uses the following commutative cryptosystem:

$$E_k(x) = x^k \bmod p$$

where k is a secret key known only to one of the players and p is a large public prime.

The above cryptosystem allows cards to be encrypted by players in a certain order and then be decrypted in the reverse order, because

$$E_A(E_B(x)) = E_B(E_A(x))$$

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

In the initialization step, the two players agree on a large prime p. Then A and B choose each their secret key k_A and k_B, respectively.

The game starts as follows:

1. A and B choose 52 values $\{x_1, \ldots, x_{52}\}$ which represent the cards in the deck.

2. B encrypts each x_i with her private key k_B;

$$c_i = x_i^{k_B} \bmod p, \forall i \in \{1, \ldots, 52\}$$

Next, B permutes the cryptograms c_i and sends the permuted c_i's to A.

3. A chooses five c_i and sends them to B. These will be the cards of B. A next chooses five more c_i, uses k_A to encrypt them and obtains

$$d_i = c_i^{k_A}, \forall i \in \{1, \ldots, 5\}$$

where, to keep notation simple, we have assumed that the first five c_i's are chosen the second time. Next, A permutes the d_i's and sends them to B.

4. B decrypts d_i with k_B^{-1}, where $k_B \cdot k_B^{-1} \bmod p-1 = 1$, to get

$$e_i = d_i^{k_B^{-1}}, \forall i \in \{1, \ldots, 5\}$$

The five decrypted e_i's are sent to A.

5. A decrypts the cryptograms e_i with k_A^{-1}, where $k_A \cdot k_A^{-1} \bmod p - 1 = 1$, and obtains her five cards.

In order for A and B to obtain more cards in subsequent rounds, the same protocol above can be followed with the cards remaining in the deck (rather than with 52).

2.2. Zhao *et al.* mental poker

Protocol [Zhao03] allows more than two players and still does not require a TTP. The protocol uses ElGamal's asymmetric cryptosystem [ElGa85], with the condition that all players use the same large prime p (the prime determining the field $\mathbf{Z}_p$ where operations are carried out). It is easy to see that, used in the above fashion, ElGamal's cryptosystem is commutative. To check for commutativity, assume that a value x is encrypted by A, then encrypted by B, then decrypted by A and finally decrypted by B:

- A chooses a random integer r_A, encrypts x with her key k_A

$$E_{k_A}(x) = (y_{1A}, y_{2A}) = (\alpha_A^{r_A} \bmod p, x\beta_A^{r_A} \bmod p)$$

and sends the result to B.

- B chooses a random integer r_B, encrypts y_{2A} with her key k_B

$$E_{k_B}(y_{2A}) = (y_{1B}, y_{2AB})(\alpha_B^{r_B} \bmod p, y_{2A}\beta_B^{r_B} \bmod p)$$
$$(\alpha_B^{r_B} \bmod p, x\beta_A^{r_A}\beta_B^{r_B} \bmod p)$$

and sends the result to A.

- A decrypts y_{2AB}:

$$D_{k_A}(y_{2AB}) = y_{2AB}(y_{1A}^{k_A})^{-1} \bmod p$$
$$= x\beta_A^{r_A}\beta_B^{r_B}(\beta_A^{r_A})^{-1} \bmod p = y_{2B}$$

and sends y_{2B} to B.

- B decrypts y_{2B} and obtains x:

$$D_{k_B}(y_{2B}) = y_{2B}(y_{1B}^{k_B})^{-1} \bmod p = x$$

Without loss of generality, we next describe the Protocol in [Zhao03] for the case of two players A and B. In the initialization step A and B agree on a large prime p which determines the finite field Z_p in which all operations will be carried out and then they compute each a secret key pair:

$$K_A = \{(p, \alpha_A, k_A, \beta_A) : \beta_A \equiv \alpha_A^{k_A} \pmod p\}$$

$$K_B = \{(p, \alpha_B, k_B, \beta_B) : \beta_B \equiv \alpha_B^{k_B} \pmod p\}$$

Now the card deck is represented by 52 values $\{x_1, \cdots, x_{52}\}$ which are agreed upon by both players. Each player encrypts the x_i's under her public key and obtains cryptograms which are sent to the other player. Then each player encrypts the received cryptograms under her public key and sends the result to the other player. As a result, both players should obtain two identical sets of cryptograms. If one or both players deviate from the protocol, the two sets of cryptograms will differ. More formally,

1. A chooses r_A and encrypts the 52 values x_i representing the cards:

$$\begin{aligned} E_{K_A}(x_1) &= (y_{1A} = \alpha_A^{r_A}, y_{2A}^1 = x_1\beta_A^{r_A}) \\ E_{K_A}(x_2) &= (y_{1A} = \alpha_A^{r_A}, y_{2A}^2 = x_2\beta_A^{r_A}) \\ &\vdots \\ E_{K_A}(x_{52}) &= (y_{1A} = \alpha_A^{r_A}, y_{2A}^{52} = x_{52}\beta_A^{r_A}) \end{aligned}$$

Now, A permutes the encrypted cards above and sends the set C_A of permuted encrypted cards to B.

2. Simultaneously, B chooses r_B and encrypts the 52 values x_i representing the cards

$$\begin{aligned} E_{K_B}(x_1) &= (y_{1B} = \alpha_B^{r_B}, y_{2B}^1 = x_1\beta_B^{r_B}) \\ E_{K_B}(x_2) &= (y_{1B} = \alpha_B^{r_B}, y_{2B}^2 = x_2\beta_B^{r_B}) \\ &\vdots \\ E_{K_B}(x_{52}) &= (y_{1B} = \alpha_B^{r_B}, y_{2B}^{52} = x_{52}\beta_B^{r_B}) \end{aligned}$$

IEEE COMPUTER SOCIETY

Now B permutes the encrypted cards above and sends the resulting set C_B of permuted encrypted cards to A.

3. When A receives the set C_B, she encrypts each cryptogram in the set by using her key K_A:

$$\begin{aligned} E_{K_A}(y^k_{2B}) &= (y_{1A} = \alpha^{r_A}_A, y^k_{2BA} = x_k \beta^{r_B}_B \beta^{r_A}_A) \\ E_{K_A}(y^l_{2B}) &= (y_{1A} = \alpha^{r_A}_A, y^l_{2BA} = x_l \beta^{r_B}_B \beta^{r_A}_A) \\ &\vdots \\ E_{K_A}(y^m_{2B}) &= (y_{1A} = \alpha^{r_A}_A, y^m_{2BA} = x_m \beta^{r_B}_B \beta^{r_A}_A) \end{aligned}$$

Call C_{BA} the set of values resulting from the above encryptions.

4. On her side, B encrypts the cryptograms in C_A under her key K_B:

$$\begin{aligned} E_{K_B}(y^r_{2A}) &= (y_{1B} = \alpha^{r_B}_B, y^r_{2AB} = x_r \beta^{r_A}_A \beta^{r_B}_B) \\ E_{K_B}(y^s_{2A}) &= (y_{1B} = \alpha^{r_B}_B, y^s_{2AB} = x_s \beta^{r_A}_A \beta^{r_B}_B) \\ &\vdots \\ E_{K_B}(y^t_{2A}) &= (y_{1B} = \alpha^{r_B}_B, y^t_{2AB} = x_t \beta^{r_A}_A \beta^{r_B}_B) \end{aligned}$$

Call C_{AB} the set of values resulting from the above encryptions.

By publishing C_{AB} and C_{BA}, players A and B *can* check that both sets contain the same cryptograms, although probably in different order.

The protocol can naturally be extended to N players. Player i encrypts the values $\{x_1, \ldots, x_5 2\}$ representing the cards with her key K_i and a random number r_i and publishes the set

$$C_i = \{c_1 = E_{K_i}(x_1), \ldots, c_{52} = E_{K_i}(x_5 2)\}$$

The $N-1$ remaining players successively encrypt (one after the other) this set C_i, to obtain

$$\begin{aligned} C^N_i &= \{E_{K_1}(E_{K_2}(\ldots E_{K_N}(x_1))), \\ &\quad \ldots, E_{K_1}(E_{K_2}(\ldots E_{K_N}(x_5 2)))\} \end{aligned}$$

The operation described for player i is also performed for the remaining $N-1$ players. Eventually, N sets $C^N_1, \cdots, C^N_N$ are obtained which are equivalent, because the cryptosystem is commutative.

To start the game, the players must agree on a common ordering of the cryptograms representing the cards. When a player wants to obtain a card, she chooses one of the cryptograms that have not previously been selected (by her or by the other players).

The 52 values $\{x_1, \cdots, x_{52}\}$ represent the card deck and they are encrypted under the private keys of all N players, that is $c_i = E_{K_1}(E_{K_2}(\ldots E_{K_N}(x_i)))$. So, when a player i chooses a cryptogram c_i, cooperation of the remaining $N-1$ players is required to decrypt c_i, *i.e.* to obtain $d_i = E_{K_i}(x_i) = D_{K_1}(D_{K_2}(\ldots D_{K_{i-1}}(D_{K_{i+1}}(\ldots D_{K_n}(c_i)))))$. Note that the cryptosystem is commutative, so that the order used in decryption is not relevant.

This decrypted cryptogram d_i does not give the other players any indication on the card x_i to which it corresponds, because x_i is still encrypted under the key of the requesting player.

3. The attack

The above presented protocols are computationally efficient and fast. Unfortunately, the use by Zhao *et al.* of an ElGamal-like commutative cryptosystem introduces a basic weakness. With little computation, a player can decrypt the encrypted cryptograms and find the clear cards x_i, for $i = 1$ to 52.

First of all, note that the attack scenario is a known-cleartext one. The cleartext card values x_i, for $i = 1, \cdots, 52$ 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 clear card x_i in it because this value is multiplied by a *hiding factor*, which will be denoted by f_h. Now in the cryptograms y^i_{2AB} or y^i_{2BA} described in the previous section all values x_i are hidden by the same hiding factor

$$f_h = \beta^{r_A}_A \beta^{r_B}_B$$

If a player multiplies any encrypted card y^i_{2AB} by f^{-1}_h, she will be able to find its value x_i without knowing the encryption key:

$$y^i_{2AB} \cdot f^{-1}_h = x_i (\beta^{r_A}_A \beta^{r_B}_B)(\beta^{r_A}_A \beta^{r_B}_B)^{-1} = x_i$$

Now it remains to show how a player can find and invert f_h:

1. Let x_{i_1} and x_{i_1} ($i_1 \neq i_2$) be two card values chosen during the initialization step.

2. The attacker computes the multiplicative inverses $x^{-1}_{i_1}$ and $x^{-1}_{i_2}$ modulo p.

3. Then the attacker multiplies each encrypted card y^j_{2AB}, for $j = 1, \cdots, 52$ by $x^{-1}_{i_1}$ and a first set of values D_1 is obtained. One of the values in this set is f_h:

$$\begin{aligned} y^{i_1}_{2AB} \cdot x^{-1}_{i_1} &= x_{i_1} \beta^{r_A}_A \beta^{r_B}_B x^{-1}_{i_1} \\ &= \beta^{r_A}_A \beta^{r_B}_B = f_h \end{aligned}$$

4. The procedure of the previous step is repeated by the attacker with $x^{-1}_{i_2}$ and a second set of values D_2 is obtained. Again, one of the values in D_2 is f_h:

$$y^{i_2}_{2AB} \cdot x^{-1}_{i_2} = x_{i_2} \beta^{r_A}_A \beta^{r_B}_B x^{-1}_{i_2}$$

IEEE COMPUTER SOCIETY

$$= \beta_A^{r_A} \beta_B^{r_B} = f_h$$

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.

4. Conclusions

We have shown that replacing an exponential commutative cryptosystem with an ElGamal-like commutative cryptosystem turns a safe TTP-free mental poker protocol (Shamir-Rivest-Adleman) into a weak protocol (Zhao *et al.*). The weakness comes from the fact that any player (or any outsider knowing the deck coding) can decrypt encrypted cards without knowing the encrypted key.

Acknowledgments

This work was partly supported by the Spanish Ministry of Science and Technology and the European FEDER Fund through project no. TIC-2001-0633-C03-01 "STREAMOBILE".

References

[Cast03] J. Castellà-Roca, J. Domingo-Ferrer, A. Riera and J. Borrell, "Practical mental poker without a TTP based on homomorphic encryption", in *Progress in Cryptology-INDOCRYPT'2003*, LNCS (to appear, 2003).

[Chou02] J.-S. Chou and Y.-S. Yeh, "Mental poker game based on a bit commitment scheme through a network", *Computer Networks*, vol. 38, pp. 247-255, 2002.

[Crép86] C. Crépeau, "A secure poker protocol that minimizes the effect of player coalitions", in *Advances in Cryptology-CRYPTO'85*, LNCS 218, Springer-Verlag, pp. 73-86, 1986.

[Crép87] 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, Springer-Verlag, pp. 239-250, 1987.

[ElGa85] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transaction on Information Theory*, vol. 31, pp. 469-472, 1985.

[Gold82] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information", in *Proceedings of the 18th ACM Symposium on the Theory of Computing*, pp. 270-299, 1982.

[Hall97] C. Hall and B. Schneier, "Remote electronic gambling", in *13th ACM Annual Computer Security Applications Conference*, pp. 227-230, 1997.

[Harn00] L. Harn, H.-Y. Lin and G. Gong, "Bounded-to-unbounded poker game", *Electronics Letters*, vol. 36, no. 3, pp. 214-215, 2000.

[Kuro97] K. Kurosawa, Y. Katayama and W. Ogata, "Reshuffable and laziness tolerant mental card game protocol", *TIEICE: IEICE Transactions on Communications/Information and Systems*, vol. E00-A, 1997.

[Sham81] A. Shamir, R. Rivest and L. Adleman, "Mental poker", *Mathematical Gardner*, pp. 37-43, 1981.

[Zhao03] W. Zhao, V. Vadaharajan and Y. Mu, "A secure mental poker protocol over the Internet", in *Proc. Australasian Information Security Workshop*, Adelaide, Australia. Conferences in Research and Practice in Information Technology, Australian Computer Society, vol. 21, (eds. C. Johnson, P. Montague and C. Steketee), pp. 105-109, 2003. `http://portal.acm.org/citation.cfm?id=828000&jmp=abstract&dl=GUIDE&dl=ACM`

IEEE COMPUTER SOCIETY