Simple TTP-free Mental Poker protocol

Choongmin Lee

Dept. of Computer Science and Engineering, Seoul National University

Received June 13, 2014; revised September 24, 2014

Abstract

In this paper, we present a protocol for playing card games fairly between potentially dishonest players without a trusted third party (TTP), typically called Mental Poker in the literature. The protocol is a variation of the protocol proposed by Shamir, Rivest, and Adleman (SRA) but uses elliptic curve point multiplication instead of modular exponentiation to encrypt and decrypt messages. Compared to the SRA protocol, our protocol does not have the security flaws pointed by Lipton and Coppersmith that leak partial information under the assumption that the underlying elliptic curve cryptosystem is semantically secure. It is also simpler to implement than most protocols in the literature.

1 Introduction

Mental Poker is a card game analogue of mental chess, played verbally or over the telephone with no actual physical objects. In a computer sense, it is a problem of playing a fair game of poker with potentially dishonest players without a trusted moderator or server, commonly called a trusted third party (TTP). Robert W. Floyd is credited as the first to question if a fair game of Mental Poker is possible [13].

Mental Poker has played an important role in the history of cryptography. For example, Goldwasser and Micali [10] invented the idea of "semantic security" to solve the Metal Poker problem in 1982. Mental Poker can be viewed as an application of secure multi-party computation. Advances in Mental Poker may help to understand and make improvements for other related subjects such as online gambling, electronic voting, and digital currency.

2 Related works

Shamir, Rivest, and Adleman [13] proposed a protocol for Mental Poker by two players that requires a commutative cryptosystem. However, their specific choice of the cryptosystem, based on modular exponentiation, was found to be insecure by Lipton [12] and Coppersmith [7], in that a dishonest player can obtain partial information about the opponent's hand if the protocol parameters are not chosen carefully. Observing this, Crépeau [8] suggested that the following requirements should be satisfied for Mental Poker:

- Uniqueness of cards
- Uniform random distribution of cards
- Absence of trusted third party
- Cheating detection with a very high probability
- Complete confidentiality of cards
- Minimal effect of coalitions
- Complete confidentiality of strategy

The protocols based on the probabilistic encryption by Goldwasser et al. [10, 11] do not leak partial information, but they are theoretical and too inefficient to be used in practice in terms of both computation time and message size. Crépeau [8, 9] proposed two protocols, first based on the probabilistic encryption of Goldwasser et al. and a year later an improved one based on zero-knowledge proofs. But these protocols are also not so efficient to be used for real games.

The protocol proposed by Bárány and Füredi [1] was unique in that it does not require a commutative cryptosystem and relies on instead exchanging permutations between players in a certain way. However, even a small coalition can break the game in this protocol: two players P_i and Q can see all other players' hands if they collude.

Zhao, Varadharajan, and Mu [18] proposed a protocol based on the SRA protocol using the ElGamal cryptosystem, but it was soon found not to satisfy the basic security requirement: it is possible for any player to sneak the hand of everyone else [3]. Zhao and Varadharajan [17] came up with a revised protocol later but it was also found flawed [5].

The protocols proposed in [2, 4, 6, 14, 15, 16] are fundamentally different from the SRA protocol and more or less complex to implement.

Our contribution is a simple protocol based on the SRA protocol. Since Lipton and Coppersmith found the SRA protocol to be insecure, it has received less attention than it deserves. However, the SRA protocol can be made secure if a proper semantically secure cryptosystem is used. Our protocol satisfies all the requirements suggested by Crépeau, including the confidentiality of strategy, under the assumption that the elliptic curve point multiplication is semantically secure. The protocol is presented in Section 3. Section 4 gives an analysis on the protocol, verifying whether it meets all the requirements of Crépeau and discussing the performance. Section 5 is a conclusion.

3 Protocol

Let the number of cards be M (typically $M = 52$). Cards are represented as numbers from 1 to M. The number of players is N ($N \geq 2$). Players are denoted as $\mathcal{P}_1, \ldots, \mathcal{P}_N$ and connected with one another in a network, forming a complete graph so that there is a communication channel between every pair of players. It is assumed that the communication channels between players are encrypted and authenticated using some kind of public key cryptography.

Domain parameters for an elliptic curve are agreed on by players in advance. These parameters should be chosen from the standard curves such as the NIST

curves so that the following cryptosystem is semantically secure. The order of the base point G, r, must be prime and large ($r > 2^{128}$). Encryption and decryption of a point P ($P \in \langle G \rangle \setminus \{\mathcal{O}\}$) on the elliptic curve with a secret key s ($1 \leq s < r$) is done as follows:

$$\begin{aligned} E_s(P) &= sP = C \\ D_s(C) &= s^{-1}C = s^{-1}sP = P \end{aligned} \tag{1}$$

where s^{-1} is a multiplicative inverse of s in $\mathbb{Z}_r^*$, or equivalently, $ss^{-1} \equiv 1 \mod r$. It works because r is prime and therefore the order of any point in $\langle G \rangle$ except $\mathcal{O}$ is also r. It is easy to see that this cryptosystem is commutative, i.e. $E_s(E_t(P)) = E_t(E_s(P))$.

Our protocol is comprised of three subprotocols: shuffling, drawing, opening. At the start of a game, the deck is shuffled once, and any number of drawing and opening operations may follow.

3.1 Shuffling

Shuffling is divided into three phases: points generation, cascaded shuffling, and locking.

3.1.1 Points generation

1. Each player $\mathcal{P}_i$ generates M random integers $m_{i,1}, \ldots, m_{i,M} \in \mathbb{Z}_r^*$ and broadcasts $\{m_{i,1}G, \ldots, m_{i,M}G\}$ using a commitment scheme.
2. After receiving the broadcasts from all players, each player $\mathcal{P}_i$ computes M points $\{P_1, \ldots, P_M\}$ where $P_j = (m_{1,j} + \cdots + m_{N,j})G$.

P_j represents the j-th card. If $P_x = P_y$ for some x, y where $x \neq y$, this phase is repeated until there is no duplicates, although the probability of two or more points having the same values is virtually zero (much less than 2^{-100} if $r > 2^{128}$).

The commitment scheme is required to prevent malicious players from manipulating the generated values to cheat. It can be as simple as first broadcasting the cryptographic hash of the data to be sent and broadcasting the data only after receiving the hashes from every player.

3.1.2 Cascaded shuffling

1. $\mathcal{P}_1$ shuffles cards $\{P_1, \ldots, P_M\}$ with a random permutation π_1 and encrypts them with a random integer $s_1 \in \mathbb{Z}_r^*$ to produce $\{s_1 P_{\pi_1^{-1}(1)}, \ldots, s_1 P_{\pi_1^{-1}(M)}\}$. $\mathcal{P}_1$ passes the results to $\mathcal{P}_2$.
2. After receiving the results from $\mathcal{P}_{i-1}$ ($i \geq 2$), $\mathcal{P}_i$ shuffles them with a random permutation π_i and encrypts them with a random integer $s_i \in \mathbb{Z}_r^*$ to produce $\{\hat{s}_i P_{\hat{\pi}_i^{-1}(1)}, \ldots, \hat{s}_i P_{\hat{\pi}_i^{-1}(M)}\}$ where $\hat{s}_i = s_1 \ldots s_i$ and $\hat{\pi}_i(\cdot) = \pi_i(\ldots(\pi_1(\cdot)))$. $\mathcal{P}_i$ passes the results to $\mathcal{P}_{i+1}$ if $i \leq N-1$ or $\mathcal{P}_1$ if $i = N$.
3. At the end, $\mathcal{P}_1$ receives $\{S_1, \ldots, S_M\}$ from $\mathcal{P}_N$, where $S_k = \hat{s}_N P_{\pi^{-1}(k)}$ and $\pi = \hat{\pi}_N$.

3.1.3 Locking

1. $\mathcal{P}_1$ decrypts $\{S_1, \ldots, S_M\}$ with s_1 and encrypts each S_k with a random integer $s_{1,k} \in \mathbb{Z}_r^*$ to produce $\{(s_{1,1}s_1^{-1})S_1, \ldots, (s_{1,M}s_1^{-1})S_M\}$. $\mathcal{P}_1$ passes the results to $\mathcal{P}_2$.
2. After receiving the results from $\mathcal{P}_{i-1}$ ($i \geq 2$), $\mathcal{P}_i$ decrypts them with s_i and encrypts each S_k with a random integer $s_{i,k} \in \mathbb{Z}_r^*$ to produce $\{(\hat{s}_{i,1}\hat{s}_i^{-1})S_1, \ldots, (\hat{s}_{i,M}\hat{s}_i^{-1})S_M\}$ where $\hat{s}_{i,k} = s_{1,k} \ldots s_{i,k}$. $\mathcal{P}_i$ passes the results to $\mathcal{P}_{i+1}$ if $i \leq N-1$ or broadcasts them if $i = N$.
3. At the end, every player has $\{C_1, \ldots, C_M\}$ where $C_k = (\hat{s}_{N,k}\hat{s}_N^{-1})S_k = (\hat{s}_{N,k}\hat{s}_N^{-1})\hat{s}_N P_{\pi^{-1}(k)} = \hat{s}_{N,k}P_{\pi^{-1}(k)}$.

The sequence $\{C_1, \ldots, C_M\}$ is called as encrypted cards.

3.2 Drawing

Suppose a player $\mathcal{P}_i$ needs to draw a card. The drawing procedure is as follows:

1. $\mathcal{P}_i$ chooses an unowned card C_k from $\{C_1, \ldots, C_M\}$ and broadcasts k.
2. Each player $\mathcal{P}_{i'}$ marks that C_k is owned by $\mathcal{P}_i$ and sends $s_{i',k}$ to $\mathcal{P}_i$.
3. $\mathcal{P}_i$ decrypts C_k with $s_{1,k}, \ldots, s_{N,k}$ to get $P_j = \hat{s}_{N,k}^{-1}C_k = \hat{s}_{N,k}^{-1}\hat{s}_{N,k}P_{\pi^{-1}(k)} = P_{\pi^{-1}(k)}$.

3.3 Opening

When a player $\mathcal{P}_i$ wants to open his card P_j, which is a decryption of C_k, it is done as follows:

1. $\mathcal{P}_i$ broadcasts k. $\mathcal{P}_i$ must own C_k to do this.
2. Each player $\mathcal{P}_{i'}$ broadcasts $s_{i',k}$.
3. Each player $\mathcal{P}_{i'}$ decrypts C_k with $s_{1,k}, \ldots, s_{N,k}$ to get P_j.

4 Analysis

4.1 Uniqueness of cards

Observe that the points generation phase in the shuffling protocol ensures that there is no duplicate points. The sequence of encrypted cards $\{C_1, \ldots, C_M\}$ is not necessarily duplicate-free, but it can be made so by appending the indices to each element to make an augmented set: $\{(C_1, 1), \ldots, (C_M, M)\}$. There is a bijection between the augmented sequence of encrypted cards and the points representing cards, $(C_k, k) \leftrightarrow P_j$ where $k = \pi(j)$. As (C_k, k) is chosen no more than once, we can conclude that every card is unique.

4.2 Uniform random distribution of cards

If there is at least one player that is honest and chooses a permutation randomly and uniformly, the resulting combined permutation π is uniformly random and so is drawing a card no matter how k is chosen when choosing an encrypted card C_k.

4.3 Absence of trusted third party

There is no one in our protocol that must be trusted by all other players.

4.4 Cheating detection with a very high probability

As the confidentiality of cards and coalitions have their own sections, let us focus on the other aspect of cheating, namely, forging cards. Is it possible for a player to forge a card, e.g. make other players believe that C_k is an encrypted card of $P_{j'}$, though it is in fact of P_j ($j \neq j'$)? Suppose that $k = \pi(j)$, $P_j = aG$, $P_{j'} = bG$, $a \neq b$. Then

$$\begin{aligned} s_{1,k}^{-1} \dots s_{N,k}^{-1} C_k &= aG \\ ba^{-1} s_{1,k}^{-1} \dots s_{N,k}^{-1} C_k &= bG \end{aligned} \tag{2}$$

If a malicious player $\mathcal{P}_i$ broadcasts $s_{i,k}ab^{-1}$ instead of $s_{i,k}$, other players can be fooled into believing that C_k is an encrypted card of the card $P_{j'}$. But finding the values of a and b requires the player to solve the elliptic curve discrete logarithm problem (ECDLP), which is believed to be infeasible, if the curve parameters are chosen carefully. If a malicious player broadcasts a random value instead of $s_{i,k}$, there is practically no chance that the decrypted value would match one of the generated points. So the answer is no; it is not possible for a polynomial-time adversary to forge a card with a non-negligible probability under the assumption that ECDLP is hard for the chosen curve parameters.

4.5 Complete confidentiality of cards

The confidentiality of cards relies on the assumption that our elliptic curve cryptosystem is semantically secure for random messages. Informally speaking, it means no information should be leaked to the ciphertext. It is not too hard to see that the elliptic curve multiplication is semantically secure if the decisional Diffie-Hellman (DDH) assumption is hold for the underlying elliptic curve. Sticking to the standard curves such as NIST curves should achieve the desired security.

If a point is a linear combination of some other points, then one can easily find which encrypted cards are for those points by trying the same linear combination on encrypted points. For example, if one finds that $P_x = P_y + P_z$ for some distinct cards, then he can try all $52 \times 51 \times 50 \times \frac{1}{2} = 66300$ pairs of encrypted cards (assuming $M = 52$) to identify those cards. However, as the points are jointly and randomly generated, no one can easily find such linear combination given just points P_j, without all the values of $m_{1,j}, \dots, m_{N,j}$, which requires solving ECDLP on the values exchanged at the first step of the points generation phase.

As long as r is sufficiently large, and considering the relatively short playing time of one round of a game, it should be infeasible to reveal which card an encrypted card represents.

4.6 Minimal effect of coalitions

The effect of coalitions is minimal, because if there is at least one honest player, the shuffling would be random. Also, all players must cooperate to decrypt an encrypted

card C_k because all $s_{1,k}, \ldots, s_{N,k}$ are needed to decrypt the card. A coalition, even if it is of the maximum size ($N-1$ out of N players), has neither information about the other players' hands nor the ability to forge cards. In other words, a coalition has no advantage over honest players except that players in the coalition can share their own hands.

4.7 Complete confidentiality of strategy

Because no one can forge a card, it is unnecessary to reveal all the secrets that were used to encrypt cards after the game. Every opening of a card is a proof of the possession of the card in itself.

4.8 Performance

When shuffling the deck, each player performs $3M$ elliptic curve point multiplications and $M(N-1)$ point additions, and transfers $M(N+1)$ encrypted points, except the last player $\mathcal{P}_N$ who transfers $M(2N-1)$ encrypted points. When drawing and opening a card, $N-1$ secret keys are transferred and one point multiplication is performed per player. When $M = 52$ and $N = 4$, that is at most 208 point multiplications, 156 point additions, 260 point transfers (364 for the last player), and 156 secret key transfers for each player. The number of operations are small, and even without point compression, the data transfer is only about 20 KB per player per game when $r \sim 2^{192}$.

5 Conclusion

A simple, complete protocol for Mental Poker that complies with all the requirements suggested by Crépeau is presented in this paper. It can be thought as an extension of the SRA protocol to three or more players with elliptic curve cryptography as the underlying commutative cryptosystem. Future research should be focused on developing formal proofs that the elliptic curve point multiplication is indeed semantically secure, because it may be the case that it leaks a few bits of information, just like modular exponentiation does. Furthermore, it would be interesting to see if this protocol can be modified to have drop-out tolerance, which is another desirable property of Mental Poker proposed by Castellà-Roca et al. [6], and more operations than drawing and opening, such as returning drawn cards to the deck and shuffling the deck in the middle of a game.

References

[1] Imre Bárány and Zoltán Füredi. Mental poker with three or more players. *Information and Control*, 59(1-3):84–93, January 1983.

[2] Adam Barnett and Nigel P Smart. Mental Poker Revisited. In *Cryptography and Coding, 9th IMA International Conference*, LNCS 2898, pages 370–383. Springer Berlin Heidelberg, 2003.

[3] Jordi Castellà-Roca and Josep Domingo-Ferrer. On the security of an efficient TTP-free mental poker protocol. In *International Conference on Information Technology: Coding and Computing (ITCC '04)*, volume 2, pages 781–784. IEEE, 2004.

[4] Jordi Castellà-Roca, Josep Domingo-Ferrer, Andreu Riera, and Joan Borrell. Practical Mental Poker Without a TTP Based on Homomorphic Encryption. In *International Conference on Cryptology in India*, LNCS 2904, pages 280–294. Springer Berlin Heidelberg, 2003.

[5] Jordi Castellà-Roca, Josep Domingo-Ferrer, and Francesc Sebé. On the Security of a Repaired Mental Poker Protocol. In *Third International Conference on Information Technology: New Generations (ITNG '06)*, pages 664–668. IEEE, 2006.

[6] Jordi Castellà-Roca, Francesc Sebé, and Josep Domingo-Ferrer. Dropout-Tolerant TTP-Free Mental Poker. In *Trust, Privacy, and Security in Digital Business*, LNCS 3592, pages 30–40. Springer Berlin Heidelberg, 2005.

[7] Don Coppersmith. Cheating at Mental Poker. In *Advances in Cryptography - CRYPTO '85*, LNCS 218, pages 104–107. Springer Berlin Heidelberg, 1986.

[8] Claude Crépeau. A Secure Poker Protocol that Minimizes the Effect of Player Coalitions. In *Advances in Cryptography - CRYPTO '85*, LNCS 218, pages 73–86. Springer Berlin Heidelberg, 1986.

[9] Claude Crépeau. A zero-knowledge Poker protocol that achieves confidentiality of the players ' strategy or How to achieve an electronic Poker face. In *Advances in Cryptography - CRYPTO '86*, LNCS 263, pages 239–247, 1987.

[10] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing - STOC '82*, pages 365–377, New York, New York, USA, 1982. ACM Press.

[11] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[12] Richard J Lipton. How to Cheat at Mental Poker. In *Proceedings of the AMS Short Course on Cryptology*. AMS, 1981.

[13] Adi Shamir, Ronald L Rivest, and Leonard M Adleman. Mental Poker. In *The Mathmatical Gardner*, pages 37–43. Prindle, Weber & Schmidt, Boston, Massachusetts, USA, 1981.

[14] Tzer-jen Wei. Communication efficient shuffle for mental poker protocols. *Information Sciences*, 181(22):5053–5066, November 2011.

[15] Tzer-jen Wei. Secure and practical constant round mental poker. *Information Sciences*, 273:352–386, July 2014.

[16] Tzer-jen Wei and LC Wang. A fast mental poker protocol. *Journal of Mathematical Cryptology*, 6(1):39–68, January 2012.

[17] Weiliang Zhao and Vijay Varadharajan. Efficient TTP-free mental poker protocols. In *International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume 1, pages 745–750. IEEE, 2005.

[18] Weiliang Zhao, Vijay Varadharajan, and Yi Mu. A Secure Mental Poker Protocol Over The Internet. In *Australasian Information Security Workshop*, volume 21, pages 105–109, 2003.