Mental Poker

Filip Langr
Faculty of Computer and Information Science
University of Ljubljana
email: fl3781@student.uni-lj.si

1 Introduction

"Mental poker" refers to problem of playing poker game when all communication between players is accomplished using messages [1] and to protocols for solving such problems. Here we present main approches to mental poker protocols, those which require trusted third party (TTP) and those which do not. The rather trivial original protocol is described. Next, we'll present requirements for mental poker as defined by Crépeau [4] and show disadvantages of previous protocol. Then we will overview zero-knowledge proving specifically in application with discrete logarithm, which is used in another section, where more advanced TTP-free mental poker protocol, meeting Crépeau's requirements, is presented.

2 Fundamentals of Mental Poker

As already stated, mental poker aims to develop protocols for playing poker (or similar card game) when all the communication is done with messages sent via a communication channel. We basically divide mental protocols to two categories" those using trusted third party (TTP) and those which are TTP-free.

Playing mental poker game with all-handling TTP is rather trivial. Every player just communicate via secret channel with TTP which deals the cards, checks the rules and figure as a referee. However, such TTP does not have to be available or it can be expensive. Moreover, theoretically every TTP can be bribed, no TTP is 100% trustworthy. So some protocols aims to reduce the necessity and influence of TTP in the game. For example, protocol proposed in [7] uses TTP only in the very beginning of each hand. Some authors, like [4], do not suppose TTP-based protocols to be reliable because of mentioned drawbacks.

TTP-free protocols are usually more complicated than those TTP-based [3], since all communication is done among potentially dishonest players without any referee. Like in the case of TTP-based protocols, some authors do not suppose TTP-free protocols to be always fair. In [2] is justified that "Without a TTP, the fairness of card dealing in the mental poker game is uncertain".

Mental poker can be viewed as a part of multi-party computation cryptography subfield (MPC). Suppose we have a party aiming to compute a value of a function F. Every participant knows just a part of necessary input arguments. MPC provides protocols for computing F while keeping users' arguments secret. Mental poker uses that principle, since it's obviously necessary to determine the winner without revealing players' cards.

In the original paper about mental poker [1] it was proved that mental poker protocol is impossible. Basically the proof says: When Alice and Bob are playing and Bob always claims to have Straight Flush, Alice either doesn't have the possibility to check Bob's hand and so check if he's telling the truth, or Alice can check Bob's hand but in such case she can check it whenever she wants to. Neither of those two possibilities are compatible with fair poker game.

Such conclusion is true from the point of view of theory of information, however, there are, of course, methods for playing mental pokers that rely on the difficulty of inverting certain cryptographic transformations. The same paper proving the impossibility also brought the first (TTP-free) mental poker protocol for two players only. The protocol is quite straightforward:

1. Players agree on random prime p and pick their private keys $k_{A,B}$.
2. Players pick commutative cryptosystem, so that $E_{k_A}(E_{k_B}(x)) = E_{k_B}(E_{k_A}(x))$. Example of such cryptosystem is e.g. RSA using the same modulo p.
3. Bob shuffles cards, encrypt each of them with his key k_B and sends them to Alice.
4. (a) When Bob should draw a card, Alice picks one (since cards are encrypted, there is no better strategy than random choice) and sends it to Bob. Bob then decrypts it and now has his drawn random card.
   (b) When Alice should draw a card, she again randomly picks one, encrypts it with her key k_A and sends it to Bob. Bob decrypts it using his key and sends it back. Alice decrypts it with her key and now has her drawn random card.
5. The game is played, whenever a player should draw a card the previous procedure is followed. All players keep log of the game progression.
6. At the end of the game, players reveal their keys, decrypt all cards and using the the log make sure that the game proceeded according to the rules.

Now, if we compare the protocol with proof statement, Alice has access to Bob's hand, however, it is computationally impossible to determine the cards without Bob's help.

3 Crépeau's Requirements for Mental Poker Protocol

Although previous protocol is easy to understand and use, it has several serious drawbacks. In [4] Crépeau came with several requirements that a good mental poker should

have and, as we will show, our protocol from previous section does not meet several of them.

The requirements are as follows:

1. Uniqueness of cards – No card should be presented more than once in a game.
2. Uniform random distribution of cards – All random hands and decks are equally possible.
3. Absence of TTP – Protcol uses no TTP.
4. High probability of cheating detection – Based on desired security level and available computational power a security parameter is picked which defines probability of detecting cheating.
5. Confidentiality of cards – No information about cards in deck and other players' hands other than what can be observed from player's hand can be derived.
6. Minimal effect of coalitions – The only benefit of establishing a coalitions is obtaining the knowledge about my collaborators' hand. I.e. since there is one non-cheating players, nobody can learn anything about his hand or cards in the deck. (Possibly surprisingly, there are protocols like [5] where players coalitions totally ruin the fair game.)
7. Confidentiality of strategy – The players' strategies are not revealed, losing player can keep his cards secret. All the concept of bluffing is based on that, which is a crucial part of poker.

Let's challenge the proposed protocol with presented requirements. First, in [6] was proved that the protocol contains a flaw allowing a player to determine the color of opponent's card. Therefore the cards confidentiality is broken. Another important drawback is the necessity to reveal cards and so the game strategy. No serious poker player would ever play such game.

It showes out that designing mental poker protocol meeting all Crépeau's requirements which is fast enough to be useful in real-life games is probably not trivial task. Crépeau himself proposed protocol behaving according to his rules [8] but at the same time incurring great computational cost [3].

4 Zero-Knowledge Proof

Zero-knowledge proof (ZKP) is a method used for proving to a questioner by a prover a knowledge of certain information but keeping the information secret at the same time. The questioner can never make the probability that the respondent really knows the information to be one, but he can make it arbitrary close to 1. Proving consists of performing certain procedure in several rounds, the number of rounds depends on desired

security level, more rounds leads to higher probability of exposing cheater. Every round questioner challenges prover with a question which needs to be correctly answered. One of practical example of ZKP is proving the knowledge of discrete logarithm of a certain value. Here we acquaint the reader with discrete logarithm ZKP, which is used in the protocol presented in the next section.

Suppose Alice knows such x that $b^x \equiv y \pmod{p}$, p is a large prime. Values b, y, p are public, x is known only to Alice. Suppose Bob wants to know whether Alice really knows x such that $x \equiv \log_b y \pmod{p}$. At the beginning of each round, Alice picks random r, compute $C = b^r \pmod{p}$ and sends C to Bob. Bob chooses one of the following challenges:

1. Either Bob asks Alice for r. Then he checks whether $b^r \equiv C \pmod{p}$. If it does, challenge was successful.
2. Or he can ask for $(x + r) \mod (p - 1)$. Then he checks whether $b^{(x+r) \bmod (p-1)} \equiv C \cdot y \pmod{p}$. If it does, challenge was successful.

Of course, if Alice does not know the value x, she can try to cheat. She has two possibilities:

1. She can guess that Bob will ask for r. In that case, she simply picks random r and compute $C = b^r$ according to the protocol and challenge is successful. However, if her guess is wrong, Bob asks for $(x + r) \mod (p - 1)$. In that case she can not provide such number since she does not know x and so whatever she sends to Bob instead of true $(x+r) \mod (p-1)$, the check at Bob's side $b^{(x+r) \bmod (p-1)} \equiv C \cdot y \pmod{p}$ will fail.
2. On the other hand, Alice can guess that Bob will ask for $(x+r) \mod (p-1)$. In such case, she can pick random R, compute $C_{cheat} = b^R \cdot y^{-1} \pmod{p}$ and sends C_{cheat} instead of C. If Bob really asks for $(x + r) \mod (p - 1)$, Alice send him R instead. Bob then should do the check according to the protocol, which in this case is: $b^R \equiv C_{cheat} \cdot y = b^R \cdot y^{-1} \cdot y = b_R \pmod{p}$, so challenge is successful. However, if Bob asks for r, then he is asking for such r that $b^r \equiv C_{cheat} = b^R \cdot y^{-1} = b^R \cdot b^{-x} = b^{R-x} \pmod{p}$, so he is asking for $R - x$. Since Alice does not know x, she can only send a random number as r, so the check at Bob's side $b^r \equiv C_{cheat} \pmod{p}$ will fail.

Notice that in a single round dishonest Alice has probability $\frac{1}{2}$ that she will successfully answer to challenge. Therefor for s consecutive rounds, there is $\frac{1}{2^s}$ probability that cheating Alice will succeed. The s is the security parameter. Notice that the probability will never be 0, but with infinite computing power it can be theoretically made arbitrarily close to 0.

5 TTP-free protocol with Player Confidentiality

Here we present the TTP-free protocol from [3] which meets all Crépeau's requirements and it's time complexity is usable in real life games.

Rather then rewriting all protocol procedures, we will try to explain its functionality in a simple way. To overview fully-defined methods, see [3].

5.1 Initialization

First note that all messages sent by players are posted on "bulletin board". All n players agree on large primes p, q such that $p = 2q + 1$, a generator α of $G \subset \mathbb{Z}_p^*$, $< G >= q$, α is a quadratic non-residue. They also agree on security parameter s.

Every player P_i computes his private key K_i, $2 < K_i < q$, K_i is odd, and publishes his public key $\beta_i = \alpha^{K_i} \mod p$. Next, players generate random odd numbers $x_i, ..., x_{52}$ representing (a unique) cards in a deck. Finally, $\beta = \alpha^{K_1 \cdots K_n} \mod p$ is computed without revealing K_i to other players. This can be done using multi-party computation methods.

5.2 Card shuffling

Initial set of cards is $C_0 = c_{0,1}, ..., c_{0,52}$, every card is represented as a tuple $c_{0,j} = (d_{0,j}, \alpha_{0,j}) = (\alpha^{x_j} 1, \beta)$. The idea of shuffling is that every player do following steps:

1. Encrypt every card in obtained deck C_i with different random number $(d_{i,j}^{r_j}, \alpha_{i,j}^{r_j})$ mod p and permute their order in a deck. A new deck C_{i+1} is generated.

2. Player proves to the rest of players that he encrypted C_{i+1} from C_i correctly and gives C_{i+1} to next player. The proving is described below.

When all player do this procedure, final deck C_n is obtained. Note that since at least one player is honest and who keeps permutation secret, other players cannot discover correspondence between $c_{n,j} \in C_n$ and cards x_j. This guarantees the confidentiality of cards. Also if at least player uses truly random permutation, the deck of cards will be uniformly permuted. As we can see, two Crépeau's criteria are satisfied.

In a step 2., the player should prove that he computed C_{i+1} from C_i correctly. This prove is based on a ZKP technique (not using discrete logarithm though), so it uses security parameter s to increase the probability of cheating detection. First, the player computes C_i s times again, obtaining $C_{i+1,k}$. Next, the rest of players produce random sequence of s bits. Next, it is verified for all s computed decks whether C_{i+1} was well computed into $C_{i+1,k}$ or C_i was well computed into $C_{i+1,k}$. Every randomly generated bit is used to decide whether the first or the second verification should be done. In order to cheat, by taking a wrong C_{i+1} and then computing suitable C_{i+1}, k other players have to decide whether first or second verification will be done. The decision is done by random using random bits, so there is $\frac{1}{2^s}$ probability that potential cheating coalition passes all verifications. This approach in general guarantee uniqueness of cards with high-enough probability adjusted by s.

5.3 Card drawing, opening and discarding

Player who wants to draw a card pick the one which was not drawn yet, suppose it is $c_{n,j} = (d_{n,j}, \alpha_{n,j})$. Next, let's take $r_0 = \alpha_{n,j}$. Every player do the following:

1. For obtained r_i computes $r_{i+1} = r_i^{K_i^{-1}} \mod p$.
2. Next, it's necessary to prove to other players that the player really uses his private key K_i in above expression. Since the key should remain private a zero-knowledge proof is used. Player is proving that $\log_\alpha \beta_i \equiv \log_{r_{i+1}} r_i \pmod p$, i.e. he is proving that the exponent in step 1. he used is really his key $K_i = \log_\alpha \beta_i \mod p$.

After $n-1$ steps, drawing player receives r_{n-1} . Now he can also compute $r_n = r_{n-1}^{K_u^{-1}}$, where K_u is player's private key. Note that initial $r_0 = \alpha_{n,j} \equiv \beta^{K_1 \cdots K_n \cdot r_1 \cdots r_n} \pmod p$, where $r_1, .., r_n$ are random exponents from card shuffling. Then the final $r_n \equiv \beta^{K_1 \cdots K_n \cdot r_1 \cdots r_n \cdot K_1^{-1} \cdots K_n^{-1}} \equiv \beta^{r_1 \cdots r_n} \pmod p$. Player now just check all 52 x_j values to find such x that $d_{n,j} \equiv r_n^x \pmod p$. This x represents the card.

In a card opening, it's just necessary to public r_n and x to prove that player really used his private key K_u to compute final r_n. This can be done again by ZKP in the same way as in the previous case. All players then can verify that $d_{n,j} \equiv e_n^x \pmod p$.

If player wants to discard the card, he sends $c_{n,j}$ to bulletin board. If any player wants to cheat and open discarded card, all players can detect the cheat because $c_{n,j}$ is on the board.

As we can see there is no TTP in the protocol. Also all possibly vulnerable parts of protocol (like card shuffling) involves all players in a way that if at least one player is honest, the procedure can't be cheated. Uniqueness of cards, uniform random distribution of cards and confidentiality of cards was already discussed in previous text. Last but not least, there is no card revealing in the end of the hand, so confidentiality of strategy is guaranteed and protocol allows players to bluff. Practical evaluation [3] shows reasonable time consumption of proposed protocol.

One of possible drawback of the protocol is that it does not support dropouts. As a result either intentional dropout (rage-quitting player) or unintentional (sudden lost of connection to communication channel) results in breaking the game since other players does not know the private key of the lost player. Protocols supporting dropout can be viewed for example here [10] [11].

6 Conclusion

In the project mental poker basics were described. Two protocols, one rather simple and one rather more complicated using zero-knowledge proving were showed and compared. There are many mental poker protocols with various features e.g. those allowing player to dropout without ruining the game, to mention one.

References

[1] A. Shamir and R. L. Rivest and L. M. Adleman, *Mental poker*, Mathematical Gardner, 1981.

[2] J-S. Chou and Y-S. Yeh, *Mental poker game based on a bit commitment scheme through network*, Computer Networks, 38(2):247-255, 2002

[3] J. C. Roca, *Contributions to Mental Poker*, Phd thesis, Universitat Autonoma de Barcelona, 2005

[4] A. Crépeau, *A secure poker protocol that minimizes the effect of player coalitions*, Advances in Cryptology, 1985.

[5] I. Barany and Z. Furedi, *Mental Poker with Three or More Players*, Advances in Cryptology, 1983.

[6] R. Lipton, *How to Cheat at Mental Poker*, Proceedings of the AMS Short Course in Cryptography, 1981.

[7] S. Fortune and M. Merritts, *Poker Protocols*, Advances in Cryptology, 1985.

[8] A. Crépeau, *A zero-knowledge poker protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face*, Advances in Cryptology, 1986.

[9] J. Castella-Roca and F. Sebé and J. Dominigo-Ferrer, *A TTP-Free Mental Poker Protocol Achieving Player Confidentiality*, Theoretial Computer Science, 2004.

[10] J. Castella-Roca and F. Sebé and J. Dominigo-Ferrer, *Dropout-Tolerant TTP-Free Mental Poker*, Trust and Privacy in Digital Business, 2005.

[11] K. Kurosawa and Y. Katayama and W. Ogata, *Reshufflable and Laziness Tolerant Mental Card Game Protocol*, TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems, 1997.