Anonymity and the Dining Cryptographers Problem

Sergey Gorbunov *Nov 19, 2012*

In this short manuscript we state the problem of sender and recipient anonymity and its connection to the dining cryptographers problem. Consider the following sender/recipient anonymity problem:

- n players wish to communicate for t rounds.
- In each round, assume only one player will speak only (this will generalized later).
- Each player P_i is given a characteristic vector v_i, where each entry $v_i[j]$ indicates whether the player should speak in round j and the message bit (for $j \in [t]$). That is, $v_i[j] = (b, m)$ where $b \in \{0, 1\}$, and if $b = 1$ then $m \in \{0, 1\}$ and otherwise $m = \perp$.

The players want to ensure the following:

1. **Correctness:** Every player should be able to recover the message sent.
2. **Non-interactiveness:** Each player broadcasts (post of bullet-in board) his message just once. There is no interaction between the players.
3. **Anonymity:** At the end of each round "no one" should be able to determine who sent the message.

Now, to satisfy **anonymity** and **correctness** only, we can use the classical multi-party computation (MPC) protocols, such as [BGW88]. In this case, we consider an OR function f that takes n inputs. The players that do not transmit give a 0 bit as their input and the transmitting player gives the message bit m as the input to the function. By the MPC security, "no one" should be able to determine the individual inputs to the function and hence the transmitting person.[1]

David Chaum proposed introduced the dining cryptographers problem, which is a fun interpretation of the anonymity problem stated above [Cha88]. We summarize the dining cryptographers problem below for the case of 3 players, which can be easily generalized.

- Three cryptographers are having are dinner at their favourite restaurant.
- At the end of the night, the waiter tells them that the dinner is paid for by one of the cryptographers of the NSA.

[1]The corruption threshold in this case is $n/2$ to obtain unconditional security. The solution we present is security for "arbitrary" corruptions.

- Respecting the privacy of each other, the cryptographers wish to determine if one of them or the NSA paid for the dinner.

Consider the following solution, which essentially gives an unconditionally security protocol for OR function.

1. Each pair of cryptographers (P_i, P_j) choose a random shared key bit k_{ij} (so, $k_{ij} = k_{ji}$).
2. Each cryptographer P_i, having neighbours P_j, P_l publishes the following:
   (a) XOR of the keys k_{ij} and k_{il} if he did *not* pay for the dinner.
   (b) Inverse of XOR of the keys k_{ij} and k_{il} if he did pay for the dinner.
3. To recover the output, each cryptographer XORs all the published messages.

Correctness. If none of the cryptographers paid for the dinner, then the output of this protocol is 0, since each key gets cancelled. Now, say P_1 for the dinner. In this case, he published $\overline{k_{12} \oplus k_{13}} = k_{12} \oplus \overline{k_{13}}$. Hence,

$$k_{12} \oplus \overline{k_{13}} \oplus k_{21} \oplus k_{23} \oplus k_{31} \oplus k_{32} = 1$$

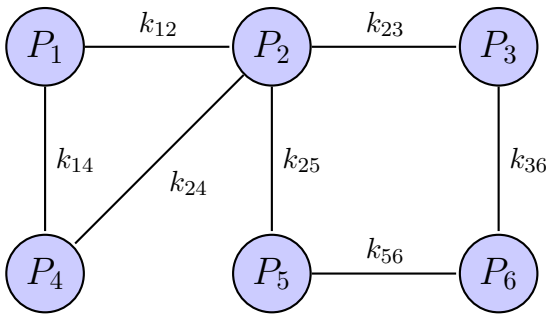
Non-interactiveness is trivial.

Anonymity. Now, say cryptographer P_1 is actually the payer. Then, we claim that P_2 will not be able to determine whether P_1 or P_3 is the payer (resp. P_3 will not be able to tell whether P_1 or P_2 is the payer). Say $k_{23} = k_{21}$ (i.e. the two coin that player P_2 sees are the same). If $k_{13} = k_{23}$ then the cryptographer who outputted 1 is the payer (since all individual XOR of the payers must be 0, and the one who inverted it must have paid). Now if $k_{13} \neq k_{23}$, then the cryptographer who outputted 0 must be payer. However, since k_{13} is randomly chosen both outcomes have equal probability and hence P_2 cannot determine who is the payer. Stated differently, the only thing that P_2 can learn is the parity of outputs of P_1 and P_3 which is $k_{13} \oplus \overline{k_{13}} = 1$ and he cant learn who actually flipped key k_{13} since he does not know what it is. The case where $k_{13} \neq k_{23}$ can be analyzed similarly.

The generalization to multiple player is trivial: each player shares a key with each other player. He then performs arithmetic over $GF(2)$ and flips the answer, if he is the payer. It is also to see how the dining cryptographers problem captures the sender/recipient anonymity problem stated above. Each player that wants to sends a message bit m, flips the answer of XOR before publishing it if $m = 1$ and does nothing otherwise. No external observer should be able to tell who sent the message.

We note that in the above protocol, each player must have pre-computed a shared secret key with every other player. However, using Diffie-Hellman key exchange this process can be performed online, obtaining computational security only.

A note on Collusions. Consider a graph $G = (V, E)$ where the players represent vertices and the edges represent the keys shared between them. Assume the graph is connected (there is a key shared between every pair of payers). Consider a collusion of m players. Clearly, if $m = n - 1$ then there is no anonymity that can be guaranteed for the remaining player. Now, if $m < n - 1$ then consider a sub-graph obtained by removing all edges (keys) known to the colluders. The remaining connected component of vertices and edges defines the anonymity set. That is, the colluders can not learn which of the remaining players sent the message. However, assume that *not* each pair of players share a key. So, the graph G is disconnected. Then, a collusion of users can potentially partition the set of remaining player and learn "something" about where the message came from. For example:



If players P_2 and P_5 collude, then the set of unknown keys form two sub-graphs: one consisting of vertices (P_1, P_4) and the other consisting of vertices (P_3, P_6). There are no keys shared between these sub-graphs. We claim that the colluders will be able to see which of the two sub-graphs the message came from. If neither P_1 nor P_4 sent the message, given $k_{12} \oplus k_{14}$ and $k_{14} \oplus k_{24}$, knowing the keys k_{14}, k_{24} the colluders can recover k_{24}. In this case, they will recover the same key k_{24} from the two broadcast they seen. Otherwise, if one of the players in this set inverted his output, the colluders will recover two different bits. Hence, they will be able to tell whether the message was sent from P_1, P_4 sub-graph.

We note that a malicious adversary (colluders) can change the outcome of this protocol just by outputting a randomly chosen bit. There are a number of subsequent works that try to address this issue [GJ04].

Finally, say the players do not have apriori knowledge about which rounds they will speak to and each player may wish to speak at more than one round.

In this case, we can modify the protocol such that each player will have n shared secret keys with each other player $(k_{ij}^1, \ldots, k_{ij}^n)$. For each $l \in n$ the player i XORs the keys he sees at position l ($b = \oplus_{j=1}^n k_{ij}^l$). He also randomly chooses $l \in n$ to which XOR his message. Of course, this protocol may introduce error.

References

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. ACM.

[Cha88] D. Chaum. The dining cryptographers problem: unconditional sender and recipient untraceability. *J. Cryptol.*, 1(1):65–75, March 1988.

[GJ04] Philippe Golle and Ari Juels. Dining cryptographers revisited. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 456–473. Springer Berlin / Heidelberg, 2004.