Advanced Topics on Privacy Enhancing Technologies
CS-523
Anonymous Communication Exercises

1 Cryptographers' Dinner

Consider a DC-networks scenario with a total of n cryptographers. Out of these, k cryptographers dislike each other and thus are guaranteed not to collude. The cryptographers decide to have a shared key setup and arrange themselves as a graph (a cryptographer is a node in the graph, edges between nodes indicate a shared key). Since a complete graph is expensive due to the large number of keys, the cryptographers form a trusted root clique structure. The structure is as follows: the k cryptographers form a root clique and share keys among themselves. All the other cryptographers create shared keys with each of the root clique members.

1. If all the members outside the root clique decide to collude, how does that affect the anonymity of the root clique?

   Solution:
   For a node not to be completely compromised, it should have at least one edge with another non-colluding node. In this scenario, members of the root clique still have edges to each other that are not compromised. The anonymity set size is k, which is the maximum possible case in this scenario.

2. If $k-1$ cryptographers finally resolve their differences and decide to collude, how does that affect the anonymity of the other nodes (assuming the other nodes do not collude with the $k-1$ nodes)?

   Solution:
   Even if $k-1$ nodes collude, there are edges between the remaining node in the root clique and every other node. The anonymity set size would be $n-(k-1)$, which is the maximum possible size in this scenario.

3. What is the total number of shared keys required in the original setup?

Solution:
The root clique members need $\binom{k}{2} = k(k-1)/2$ keys among themselves. Each of the other members has shared keys with each root clique member, leading to $k(n-k)$ keys. Hence, the total is $k(k-1)/2 + k(n-k)$ keys.

The cryptographers decide to redesign their network. They form a structure as follows: they arrange themselves in cliques of k members. Within each clique, members create shared keys with every other member. There are l cliques in total. All the cliques are then arranged in a ring structure. Every clique selects a node as a leader, and leaders of each clique share keys with their immediate neighbors in the ring.

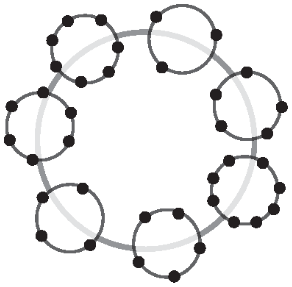

Figure 1: Topology for the second part of Question 2.

4. Within a clique, if m members decide to collude, how does that affect the anonymity of the clique?

   Solution:
   Since the clique is a complete graph, non-colluding members will have edges to other non-colluding members within the clique. The size of the anonymity set for the non-colluding members would be $k - m$, which is the maximum possible size in this scenario.

5. If two cliques decide to collude, how does that affect the anonymity of the entire setup?

   Solution:
   If two cliques collude, they can partition the ring. If the size of one partition is j cliques, the anonymity set sizes of the non-colluding members in each partition is $j * k$ and $(l - j - 2) * k$. This would be lower than the maximum possible anonymity set in this scenario (maximum possible anonymity is $(l - 2) * k$).

6. What is the total number of shared keys required in this setup?

   Solution:
   The clique members need $\binom{k}{2} = k(k-1)/2$ keys among themselves. Leaders share keys with their neighbors in the ring structure, leading to l keys. Hence, the total is $(k(k-1)/2 + 1) * l$ keys.

2 Statistical Disclosure Attack

Alice uses a mix with a total of N participants. The mix works in rounds: in each round, the mix waits for $K < N$ users to send their messages. A message can be sent to any participant in the network.

Alice uses the mix multiple times. She is wondering how likely an adversary is to figure out that she is talking to Bob, one of the people she communicates with using the mix. The adversary that Alice is concerned about can observe network traffic: He knows who participates in each round, who sends messages, and who receives the messages.

Consider the following probabilistic model of each round in which Alice participates:

- The probability that the receiver i gets a message from Alice:

$$P[i \leftarrow \mathit{Alice}] = a_i$$

- The probability that the receiver i gets a message from a sender $j \neq \mathit{Alice}$:

$$P[i \leftarrow j] = u_{ij}$$

Over T rounds in which Alice participates, the adversary keeps track of all receivers' statistics $\bar{O}_i = \frac{1}{T}\sum_{t=1}^{T} o_i^{(t)}$, where $o_i^{(t)} \geq 0$ indicates the number of messages that i received in round t. Because all rounds are independent, these observations can be thought as i.i.d samples from a random variable $O_i = \mathbf{1}_{i \leftarrow \mathit{Alice}} + \sum_{j \neq \mathit{Alice}} \mathbf{1}_{i \leftarrow j}$.

1. In terms of the probabilistic model given above, what is the expected number of messages that the receiver i gets in one round, $\mathbb{E}[O_i]$? In T rounds? **Solution**:

$$\begin{aligned}\mathbb{E}[O_i] &= \mathbb{E}[\mathbf{1}_{i \leftarrow \mathit{Alice}}] + \sum_{j \neq \mathit{Alice}} \mathbb{E}[\mathbf{1}_{i \leftarrow j}] \\ &= P(i \leftarrow \mathit{Alice}) + \sum_{j \neq \mathit{Alice}} P(i \leftarrow j) \\ &= a_i + \sum_{j \neq \mathit{Alice}} u_{ij}\end{aligned}$$

Because rounds are independent, the expected number of messages in T rounds is $T \cdot \mathbb{E}[O_i]$.

2. Assume that Alice participated in enough rounds so that the Law of Large Numbers (LLN) applies to the average statistics: $\bar{O}_i \approx \mathbb{E}[O_i]$. How can the adversary estimate a_i from a given model of u_{ij}? **Solution**:

$$a_i \approx \bar{O}_i - \sum_{j \neq \mathit{Alice}} u_{ij}$$

3. Suppose that Alice communicates *only* with Bob. The threshold of the mix is $K = 30$, and the total number of participants is $N = 290$. The adversary assumes that all users other than Alice send messages uniformly to any other user: $u_{ij} = \frac{1}{N}$. What is the expected number of messages that Bob and any other recipient receives in one round?

   Over $T = 1000$ rounds, the adversary observes that $\bar{O}_{Bob} = 0.9$. Adversary also suspects that Alice might be talking to Carol. Carol's statistic is $\bar{O}_{Carol} = 0.1$. What are the adversary's estimates for Alice's probability to send messages to Bob and to Carol?

   Solution:

$$\mathbb{E}[O_{Bob}] = a_{Bob} + \frac{K-1}{N}$$
$$\mathbb{E}[O_{Carol}] = a_{Carol} + \frac{K-1}{N}$$

   Because Alice only talks to Bob on this mix, $a_{Bob} = 1$, $\mathbb{E}[O_{Bob}] = 1 + \frac{29}{290} = 1.1$, and $\mathbb{E}[O_{Carol}] = \frac{29}{290} = 0.1$.

   The estimated probabilities of Alice sending messages are:

$$\hat{a}_{Bob} = 0.9 - \frac{K-1}{N} = 0.8, \quad \hat{a}_{Carol} = 0.1 - \frac{K-1}{N} = 0$$

   Hence, under this model of communication, the adversary can be quite sure that Alice is communicating with Bob, not Carol.

3 Getting Through a Crowd

The Crowds system may have a high latency depending on its parameters, and some extensions like "always down or up" (ADU) aim to improve this latency. In this exercise, we study the latency and privacy of these extensions.

1. A good measure for the latency of messages in crowds is the number of IP hops l. Compute the expected value and variance of the number of hops in crowds.

   Solution:
   After the first hop which does not allow direct sending, a crowds system with the send probability p_s sends the message following a geometric distribution which leads to the expected value $\mathbb{E}(l) = 1 + \dfrac{1}{p_s}$ and variance $\mathrm{var}(l) = \dfrac{1-p_s}{p_s^2}$.

In an ADU with parameters (e, l, h, m), the sender chooses a random number $u \xleftarrow{\$} [1, m]$ before sending the message. If $i \in [1, e] \cup [m-e, m]$ the sender directly sends the message. If the number is less than a lower bound $e < u \leq l$ or higher than an upper bound $h \leq u < m - e$ then the sender proceeds with AD or AU respectively. If the number is in the middle range $l < u \leq h$ then the sender chooses the direction randomly from AD and AU and forwards the mode and u with the message to the next node.

2. Compare the privacy of ADU with AD and AU.

   Solution:
   AD and AU have identical privacy guarantees due to their symmetry. After choosing the direction in the first round, ADU is identical to the AD/AU with scaled parameters based on e since every node learns the direction. Therefore, all three have the same privacy guarantee.

3. Compare the privacy of ADU with crowds.

   Solution:
   In the AD, u is always decreasing and can act as a measure of how likely is the previous node to be the sender. ADU has the same privacy as AD.

An alternative to ADU, is the "random always down or up" (RADU) algorithm is which does not send the direction (AD or AU) with the message. Each node on the path receives the message with a random number u. Similar to ADU, each node uses the ADU algorithm to decide the direction but with fresh randomness.

4. Compare the latency of ADU with RADU.

   Solution:
   RADU can change direction during the send process, so RADU has a higher latency than ADU. For example, consider the following RADU run with parameters ($e = 1, l = 1, h = 9, m = 10$). The sender chooses the random number $u_0 = 3$ and forward to the first node. The first node chooses the 'up' direction and a new random number $u_1 = 7 \xleftarrow{\$} [3, 10]$ and sends u_1 to the next node. The second node randomly chooses direction 'down' and random number $u_2 = 4 \xleftarrow{\$} [1, 7]$

5. Compare the privacy of ADU with RADU.

   Solution:
   Since the adversary does not know the direction of the previous node, it cannot use the distance of u from 1 or m as a measure of the number of hops.