CS-523: Final Exam Questions 2022

Question 1: PSI From SMC

Private Set Intersection (PSI) is a useful building block for privacy-preserving contact discovery. The PSI protocol runs between a client and a server. The client inputs their set of contacts, i.e., phone numbers, and the server the set of the phone numbers of users of the service. For simplicity, let's assume that both sets contain n items. At the end of the protocol, the client learns the intersecting items between their set and the server's set. The server learns nothing. In this question, you are tasked with building a garbled circuit-based PSI protocol.

A naïve circuit would compare each item from the client set with all items from the server set, requiring n^2 expensive circuit-based comparisons. To make the protocol more efficient, we will develop a protocol that compares each client item with a subset of m server items only ($m << n$). The key trick is to group phone numbers into bins, and then compare a client's item with the items in the corresponding server bin only. Here is how the binning works. The client and server create groups of phone numbers with the same last d digits (where d can be chosen as a function of m). For example, when considering the last $d = 2$ digits, there are 100 possible groups. The client now only needs to compare each phone number with the server's items in the corresponding bin.

Part 1: Propose an end-to-end PSI protocol that uses binning (as described above) to improve efficiency and garbled circuits to help achieve privacy. Describe your algorithm in pseudo code. Use garbled circuits as a black box. For your circuits, clearly specify the inputs, the functionality, and the outputs; here is an example (you will need another circuit for this assignment!):

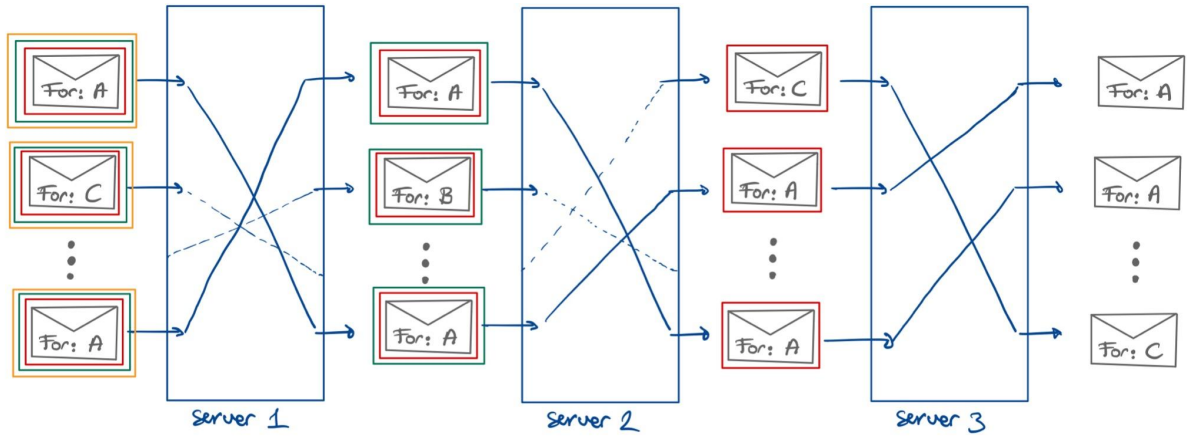
> *Alice and Bob run a garbled circuit protocol. Alice inputs her value xA, Bob inputs his value xB. The circuit computes the bit xA > xB and securely outputs the bit to Alice. Bob learns nothing.*

Part 2: Argue why this solution provides the PSI functionality, i.e., the client learns the intersecting items and nothing more, and the server learns nothing about the client's set.

Question 2: Anonymous communication using DP

One of your friends has been working on a simple system, called Piccolo, that lets users send messages to other users anonymously. Piccolo is based on a mix network operated by several servers arranged in a cascade. See the figure for an example with 3 servers. It operates in rounds. At the start of each round, the first server receives onion-encrypted messages from all users that want to send a message. The server removes one layer of encryption, shuffles all messages, and forwards them to the next server in the cascade. Each server proceeds the same: removing one layer of encryption, shuffling the result, and forwarding the messages to the next server. The final server only decrypts the messages

and obtains the destination user and the payload. It forwards the payloads to the destination users.



Throughout this assignment assume that payloads are end-to-end encrypted and have a fixed size. The final mix server thus doesn't learn anything about the content of messages.

Part 1. Consider a single round and suppose Alice receives a single message. Argue what the final server can learn about the sender of the message by colluding with all other mix servers, except for one honest server in the middle (e.g., this server is not the first nor the last in the cascade).

Part 2. You realise that this system is vulnerable to intersection attacks when the first and the last server collude: In every round, the last server observes the number of messages Alice receives. The first server reports whether Bob has sent a message. After one day, the last server counts the number of messages Alice has received over all rounds in which Bob has also sent a message. It concludes that Bob is talking to Alice if the number of such messages received is above a threshold.

You discuss this problem with a CS-523 TA: they hint that you can use differential privacy to mitigate this attack. To achieve differential privacy, the honest middle server inserts in each round multiple (onion-encrypted) dummy messages addressed to Alice into the cascade. How should the honest server determine how many messages to insert to ensure differential privacy? Assume that each round in Piccolo takes one hour.

Question 3: Anonymization / Privacy-Engineering

The Swiss government is running a competition to design a new digital distribution system for allocating study places at its two biggest, federal universities, ETH and EPFL. Study places are assigned based on applicants' final grades in their high school exams. In the final round, the competition committee is evaluating two design proposals.

System 1: Applicants upload their grades via an online portal. All grades are stored encrypted using a homomorphic encryption scheme on a central server administered by the Swiss Federal Ministry of Education. The encryption and decryption keys are stored on a

server, separate from the grades server, administered by the BSI, the Swiss Federal Office for Information Security. To allocate places, the central server runs a ranking algorithm on the encrypted grades and then asks the BSI server to decrypt the list of applicant names ranked by their final grades.

System 2: Each applicant instals an application that locally stores grades. To allocate study places, the application runs a multi-party computation protocol between all applicants which computes the ranking of the applicants. The final result, a list of applicants ranked by their final grade, gets submitted to the Swiss Federal Ministry of Education.

Part 1: Use the privacy-by-design strategies seen in class to argue which of the two system proposals, by design, achieves better privacy risk minimisation for applicants.

Part 2: Propose **one** technical privacy-enhancing modification to the system proposal that you deem **less** privacy-preserving of the two.

Question 4: Tracking

In class, you have seen various techniques to conduct *browser fingerprinting:* ways to track a user across different websites without resorting to cookies. In this question, let us think about possible client-side defences against these fingerprinting techniques. One possible defence strategy is to *detect & block*:

- **How and why does it work?** The user's browser detects scripts that are either known to conduct browser fingerprinting (e.g., through a pre-compiled block list of tracking domains or URLs), or dynamically detects suspicious behavior (e.g., a specific pattern of API accesses to a canvas object in order to detect canvas fingerprinting attempts). After detection, the browser blocks those scripts.
- **Drawbacks.** This method is only capable of preventing previously known fingerprinting methods that rely on scripts, thus suffers from two drawbacks: (1) does not cover the whole range of possible browser fingerprinting techniques (e.g., those that use images or iframes instead of scripts), and (2) is vulnerable to new script-based fingerprinting methods not seen or foreseen by the designers of the defence. Moreover, in the case of a false alarm, the defence can interfere with the legitimate functionality of the website by blocking a non-tracking script.

Another approach to preventing browser fingerprinting is modifying the browser's behavior with respect to different protocols and APIs (e.g., modifying the standard User-Agent string, or canvas behavior) in order to break the fingerprints. Following the *detect & block* example above as a template, describe **one** other defence strategy against browser fingerprinting that leverages the latter (modified protocol & API behavior) approach:

- How exactly does the defence modify the behavior and why does it prevent browser fingerprinting?
- List **at least one** potential drawback of the method in terms of either protection or utility.

Question 5: Censorship Resistance

The country Imaginia holds a state-wide municipal election. The current governing party, Party of Imaginia, managed to block any popular candidate from the opposite movement, Free Imaginia, from participating.

The opposition needs to find a way to spread information about their remaining less-known candidates. The information includes a list of all candidates (~1000) with their names, brief bio, and political views. However, the following circumstances complicate the situation:

1) The government is so powerful that they can and will try to block any internet resource they do not approve of based on the resource's IP address, unless doing so incurs serious financial losses.
2) The majority of the population are not tech-savvy.

All things considered, the opposition considers the following distribution mechanisms:

a) Use an email mailing list to publish the list to known supporters.
b) Create a website listing the new candidates, and make it available via a Tor onion/hidden service only.
c) Hire an external ISP to aid with censorship circumvention via Telex.

Part 1: For each mechanism, name one advantage and one disadvantage.
Part 2: Suggest an option "d" which does not suffer from the disadvantages you identified for the other mechanisms. Argue why that is the case.