Q1: Privacy-preserving cryptography

FLEP and ZHET, two sister universities, want to determine which of them has a smaller carbon footprint. They will determine this by computing the percentage of students coming to campus by bike for one week. To do so, each university enables their campus app to use location tracking to infer a student's mode of transportation on their way to campus. When a student enters the university campus, their app automatically reports whether they came to campus by bike or not. Then, the two universities can publish the percentage of students that go to campus by bike to determine which one is more environmentally friendly.

Part 1: Describe a Secure Multiparty Computation scheme that the apps of each university could implement to compute the percentage of students that came to campus by bike during the week. This percentage is computed over all of the students on all working days of the week (i.e., each student can use the bike one or several days of the week).

Specify:

- which circuit(s) need to be implemented,

- the input that each app gives to the circuit(s),

- how each app computes their shares,

- to whom each app sends shares,

- how the final result is computed,

- the threat model under which your solution would provide privacy guarantees, including what assumptions are needed on the capabilities of the actors in the system.

Part 2: In the scheme you described in Part 1, would it be possible for the universities to cheat to bxplain an attack. If not, justify why.

Part 3: Assume that the universities move to a Fully Homomorphic Encryption solution in which apps send their contributions to a cloud server. The server produces the aggregate statistics along with a zero-knowledge proof that it has used all the registered accounts in the system. Assume the FHE scheme and ZK proofs are implemented correctly and efficiently. Can the universities still cheat to win the competition? If yes, explain how. If not, explain what prevents cheating.

Q2: Censorship resistance & Tracking

Cameron is a journalist working at the Old Cambridge Times (OCT) who is reporting to the journal from a conflict zone. Cameron has gotten access to evidence of crimes happening in the conflict zone and needs to send some files with this evidence back to the OCT headquarters for publication.

Part 1: When Cameron tries to visit the OCT employees' website to upload the files, they discover that the website is not accessible from the conflict zone. Cameron notices that it is because the government that controls the conflict zone is censoring access to foreign websites by checking DNS requests and dropping the DNS servers' responses. A colleague tells Cameron to use the Tor network. Cameron tries, and it works! They manage to send the evidence! However, the next day the local police knocks on their door to seize their computer because they knew Cameron leaked the documents.

Part 1.1: Explain why the local government's censorship mechanism was not able to block Cameron from sending files to OCT via Tor.

Part 1.2: Explain how the police detected that Cameron sent the files using OCT's employees' website. Provide an alternative means which could enable Cameron to send the files to the OCT without getting detected by the police.

Part 2: Cameron is worried that web trackers may be learning about their activities on the Internet and giving this information to the government controlling the conflict zone. To avoid tracking, Cameron deletes all browsers from their computer and installs from scratch "Nanocroft Corner", a new privacy-preserving browser that includes three extensions that prevent tracking by blocking requests associated with tracking behaviours: AdBlocker, Ghostery, Privacy Badger.

Does the use of "Nanocroft Corner" protect Cameron from being tracked across the web? If yes, justify why tracking is not possible. If not, explain one way in which Cameron can be tracked across the web.

Q3: Machine Learning

YourRecipe is an app with which customers take a photo of their fridge, and the app returns a few recipes that they can prepare with the ingredients they have. The app uses the latest computer vision machine-learning models to recognize ingredients and then uses those ingredients to predict the most suitable recipes.

Part 1: Describe one privacy concern that arises for users of the app. Your answer must describe a threat model in which this concern would materialize.

Part 2: Compare the following two implementations in terms of how well they protect against the privacy concern you have described in part 1 and the utility for the users. Your comparison must be explicit about the threat model(s) you are assuming. This threat model can be different from the one you described in part 1.

Implementation A.

Training time: YourRecipe uses a centralized computer vision model running on a central server. This computer vision model has been trained with photos of fridges that the company's employees have voluntarily provided and labeled themselves for 1 year.

Inference time: Users send their photos to YourRecipe's server. The server queries the model and returns a list of recipes to be sent to the users.

Implementation B.

Training time: YourRecipe uses a federated approach in which users' apps collaboratively train a global model. Each user builds a local dataset that consists of photos of their fridge that they manually label. Then, the apps send a gradient computed using the local data to YourRecipe's server at each epoch. The updates received are used to improve the global model which is sent back to the users' apps.This is repeated until the global model reaches good enough performance.

Inference time: Users' photos are processed locally by the app that computes the list of recipes using their local copy of the global model and shows them to the user.

Part 3: Would your comparison change if both implementations were enhanced to perform differentially private training at instance level? If yes, explain how. If not, justify why the enhancement does not make a difference.