IEEE Services Hackathon: Problem Description

Privacy-Perserving Integration of Personal Mobility Information

1. Introduction/Context/Objectives

We can obtain our own personal daily mobility trajectories/traces using Google Location Service from Google Takeout, Integrating such information from individuals could benefit a number of use scenarios. For example, for pandemic control and precaution purposes, users can query whether they have been in the proximity of high-risk areas. In addition, such information can be used to for social recommendation, product recommendation, user profile analysis and so on. However, the privacy concerns pose significant challenges to unlocking of the hidden values in such data. The problem is simply how we can integrate such personal mobility information without disclosing anything sensitive, and in the meanwhile, the integrated set of individual mobility histories should be available for query, visualization, analytics and even model training collectively. Various technologies such as differential privacy [1, 2], encryption [3, 4], federated learning [5, 6], federated database [7], may be beneficial in addressing the problem.

2. Motivating scenarios (use cases)

We give four use scenarios of the integrated mobility data to further analyze the requirements. Your solution should be focused on one scenario, which can be one of, but not limited to any of these scenarios.

- ***Pandemic control and precaution.*** In this scenario, each individual voluntarily upload their daily mobility histories, as well as health check information, e.g. whether they have coronavirus symptoms, whether they have contacted any coronavirus patients, and whether they have diagnosed with coronavirus. Then users who have been in the proximity of high-risk locations will be notified.
- ***Product recommendation.*** In this scenario, the service provider is able to collect the mobility history of each individual. These data will be used to analyze and infer the preference of each user and recommend restaurants, coffee shops, events, and so on.
- ***User profile analysis.*** In this scenario, the service provider is able to collect mobility history of each individual and their preference information such as hobby, age, favourite books, educational level and so on. These data will be used to analyze and discover patterns of the group of users who frequently visit a certain location, which can be a restaurant, a coffee shop, an event and so on.
- **Map Service.** In this scenario, each individual voluntarily upload their daily mobility histories, and the data will be integrated with a map, so that it will visualize mobility at each location at different times. Users can also use this information to choose less popular places for shopping, travel and so on.

Datasets and Logistics: *each team should collect their own traces using* Google Location Service from Google Takeout, and anonymize the data by themselves if needed.

Privacy Requirements: **Privacy perserving is NOT required in this Hackathon Competition**, but it is recommended to provide a certain level of privacy perserving and the level of privacy preserving will be used as an important evaluation criteria.
In all of these scenarios, ideally, the service maintainer, users, and any attackers should not be able to obtain any sensitive personal information from the system by any types of attacking approaches, e.g. directly looking at the information, issuing queries to the information, sending multiple manually created trajectories, and so on. It is encouraged to carefully consider the privacy requirements. Providing a better level of privacy preserving will get better scores. The different levels of privacy preserving with some referenced technologies are provided as follows (You are not limited to these levels and technologies):

Level 1. Only the channel of uploading sensitive data is unsafe.
For example, attackers eavesdrop on the communication between two targets.

Level 2. Supposing the service provider is trustable, and users are not trustable, we need make sure no private information can be disclosed by issuing different queries through the service or sending multiple manually created trajectories, and so on.
For example, Netflix released a dataset of their user ratings as part of a competition in 2007. The dataset is anonymized, but researchers recovered 99% of all the personal information using auxiliary information from IMDB [8].

Level 3. Supposing both of the service provider and service users are not trustable, we need make sure no private information can be disclosed by looking at the data or issuing different queries through the service.
For example, we are executing software on a remote computer owned and maintained by an untrusted party. Usually Intel SGX and Homomorphic encryption can be utilized to address such problems [3].

Reference technologies:
differential privacy for deep learning, e.g., https://github.com/tensorflow/privacy
differential privacy for aggregation queries, e.g., https://github.com/google/differential-privacy
TEE and Intel-SGX, e.g., https://github.com/openenclave/openenclave
Python cryptograph libraries, e.g. https://github.com/pyca/cryptography

3. List of expected deliverables.

1. Source code that must be uploaded to an open github repository and must have a well-documented README file describing the steps of using the product;
2. Video recording of demonstration that must be uploaded to youtube;
3. A report no more than 6 pages that describes:
-- the motivation,
-- use scenario,

-- at which level privacy preserving is provided and explanation of how privacy preserving is provided,
-- technologies used,
-- novelty of the product;
The report must contain the url to the github repository and the link to the youtube video.

4. Evaluation criteria

1. Novelty of the idea (0 – 9 scale);
2. Privacy level and explanation (0 -9 scale)
3. Clarity of the report (0 – 9 scale);
4. Easiness for setting up and using the system (0 –9 scale);
5. Performance, usability, and reliability of the system (0-9 scale)

5. S/W H/W Constraints of the solution

An emulation of the application should be easily setup in one desktop through docker containers (i.e., local host service + local application emulation). The github repository should detail the steps for repeatability using docker containers.

6. Team formation

A team should have no more than six persons;

References

[1] Dwork, C. (2008, April). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation* (pp. 1-19). Springer, Berlin, Heidelberg.
[2] Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, *9*(3-4), 211-407.
[3] Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." *IACR Cryptol. ePrint Arch.* 2016, no. 86 (2016): 1-118.
[4] Gentry, Craig, Amit Sahai, and Brent Waters. "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based." *Annual Cryptology Conference*. Springer, Berlin, Heidelberg, 2013.
[5] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1-19.
[6] Konečný, Jakub, et al. "Federated learning: Strategies for improving communication efficiency." *arXiv preprint arXiv:1610.05492* (2016).
[7] Sheth, Amit P., and James A. Larson. "Federated database systems for managing distributed, heterogeneous, and autonomous databases." *ACM Computing Surveys (CSUR)* 22, no. 3 (1990): 183-236.
[8] McSherry, Frank, and Ilya Mironov. "Differentially private recommender systems: Building privacy into the netflix prize contenders." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009.