CS 208 Project Description:
A Modular System for Local Differential Privacy

Andrew Shackelford and Peter Chang

April 9, 2019

Research Questions

With this project, we aim to address several research questions.

- What is the best way to implement local differential privacy?
- Why have large tech companies only implemented local differential privacy in small instances?
- What needs must a local differential privacy framework address?
- How could already existing local differential privacy frameworks be improved so that they are easier to implement?

Local differential privacy has been touted by many tech companies as a way to truly revolutionize collecting telemetry data. Despite these claims however, the current implementations of local differential privacy are few and far between. Apple uses local differential privacy to detect new emojis, phrases, and some other small telemetry data. However, Tang et al. found concerning amounts of privacy loss over multiple analytics reports. Google uses RAPPOR to collect analytics data from Chrome, yet has not updated the software since 2016, and is currently phasing it out. Lastly, Microsoft Research published a paper and wrote a blog post about collecting telemetry data privately, but has not actually publicized the use of local differential privacy in any of its products.

As a result, the only open-source local DP product, RAPPOR, is now being phased out, and there are few if any open-source local differential privacy frameworks. While there have been many frameworks for centralized differential privacy, such as Ektelo, Psi (Ψ), and PinQ, there exist no equivalent frameworks for local differential privacy.

Therefore, we aim to design and describe a modular, extensible framework for local differential privacy, so that any interested party can easily implement local DP and know that their data collection mechanism is private. If time allows, we hope to implement a small

subset of the framework as a proof of concept, allowing for others to extend the framework to their needs. By doing so, we hope that we can increase utilization of differential privacy throughout the industry, and further protect the privacy of users.

Past Work

Google

Google developed and deployed Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR) to collect statistics through Google's Chrome Web browser.

RAPPOR employs two stages of randomized response. First, using a Bloom filter containing the client's private data, it generates a "fake Bloom filter" whose bits are determined via randomized response using the original Bloom filter's bits. This filter is memoized and permanently reused when reporting the client's particular value it represents. Next, a bit array is generated with its bit elements determined via randomized response using the bits of the fake Bloom filter from the previous step. This randomization step is recomputed for each transmission of data.

Note that since the first randomization step is not reversible via averaging over repeated observations, it ensures long-term privacy. On the other hand, the second randomization step, which is reversible via averaging, makes it very difficult for adversaries to uniquely identify a client based on the values of the permanent Bloom filter, and hence ensures short-term privacy.

Apple

Apple uses local differential privacy in several use cases in order to learn more about its users' habits, specifically estimating the frequencies of elements, without compromising privacy. Its main use cases include determining which emojis are popular, identifying websites that high resource usage or crash often, or determining which custom phrases are popular.

In contrast to Google, Apple does not employ two stages of randomized response. Instead, Apple immediately discards any identifying information, including IP address, as soon as the information is received at the server. Only then is the data ingested, aggregated, and eventually analyzed.

In order to reduce transmission costs, Apple utilizes both the Count Mean Sketch and the Hadamard Count Mean Sketch algorithms to decrease the sizes of the resulting differentially private vectors. Apple also uses the Sequence Fragment Puzzle algorithm in order to detect new phrases without having to loop through all possible combinations.

Anticipated Approach

In order to start on this project, we will look at existing frameworks, such as Apple's

analytic reports and Google's RAPPOR, and determine what features are necessary for a local differential privacy framework. We will then design an outline of our framework that would be modular and extensible for all of our desired features. Then, we will detail how our framework would incorporate these features, and what a plausible local DP pipeline could look like. Finally, if time allows, we will implement such a pipeline, or a subset thereof, so that we can test our local differential privacy framework in action.

Timeline

Below are a set of deadlines that we hope will guide our completion of this project.

- April 16: Develop an outline of the local differential privacy framework
- April 23: Finish the complete blueprint and implementation guidelines for the framework
- April 30: Finish the first part of the paper that pertains to the framework details
- May 7: Implement a subset of the framework in code, and finish the second part of the paper that pertains to the implementation
- May 10: Revise paper for clarity and submit draft
- May 13: Create presentation
- May 17: Incorporate suggestions from professors into final draft of paper

Fallback Plans

As described above, we consider our project one with two major components: theoretical blueprint and the actual implementation. If the first step proves a significantly difficult, time-consuming, and fulfilling task on its own, we would be happy to consider the implementation step a project for the future.

In the case that even the first step proves too challenging, we will limit the scope of our project to suggesting improvements for existing frameworks. For example, in class Professor Vadhan suggested that RAPPOR may perform the best when using just one hash function for its Bloom filter, and that a Bloom filter may not be the appropriate data structure at all. We could follow and extend this approach by suggesting areas of improvement and experimentally verifying them. Since RAPPOR is the only open-source local-DP framework, experimental work may only be possible with RAPPOR and the work concerning other frameworks may be purely theoretical.