Benjamin Weggenmann—Ed. (SAP)
Daniel Bernau (SAP)

Final Report

Deliverable D6.2

January 31, 2019
PANORAMIX Project, # 653497, Horizon 2020
http://www.panoramix-project.eu

Dissemination Level: Public

Horizon 2020
European Union funding
for Research & Innovation

Revision History

Revision	Date	Author(s)	Description
0.1	2018-08-01	BW, DB (SAP)	Initial draft
0.2	2018-08-16	BW, DB (SAP)	Added utility
0.3	2018-08-30	BW, DB (SAP)	Added integration with infrastructure
0.4	2018-09-03	BW, DB (SAP)	Added attacks
0.5	2018-09-10	BW, DB (SAP)	Draft for review
0.6	2018-11-29	BW (SAP)	Added utility results
0.7	2018-12-03	DB (SAP)	Added membership inference results
0.8	2018-12-12	BW (SAP)	Added religion inference results
0.9	2018-12-18	BW (SAP)	Added performance results
1.0rc	2018-12-18	BW, DB (SAP)	Final draft for review
1.1	2018-12-19	AP (UCL)	Review
1.2	2018-12-20	BW (SAP)	Revision based on review
1.3	2018-12-21	MW (UEDIN)	Final edits and draft submission to EC
1.4	2019-01-13	BW (SAP)	Added additional performance results
1.5	2019-01-15	MW (UEDIN)	Final edits and submission to EC
1.6	2019-01-31	MW (UEDIN)	Final version submitted to EC

Executive Summary

This report documents the work performed between months 24 and 41 in work package 6 of the PANORAMIX project. The goal of work package 6 (WP6) is to demonstrate the use and benefits of privacy-enhancing techniques, particularly mix networks and differential privacy, in a representative big data scenario. To that end, we develop and evaluate a prototype where sensitive data shall be collected in a privacy-friendly way from several clients to be stored and processed in a central server for further analysis. This document constitutes the final report for work package 6 and focuses on validation and testing of the developed demonstrator regarding performance, usability, anonymity, data confidentiality, as well as accuracy for the desired analysis task.

The specific use case demonstrated in WP6 focuses on taxi trip data and location data which is sent from simulated taxis and user devices over the mix network to a central aggregation server. The data will be anonymized by the taxis directly at the data source, before its transmission. This specific scenario provides a highly illustrative example of a generic data collection scenario where sensitive data should be collected in a privacy-preserving manner, requiring privacy-protection for both the transmission *metadata* and the *actual data values* themselves. In the demonstrator, we protect metadata such as source network address and timestamps using the Panoramix mix network framework. On the other hand, we protect the location data (e.g., pickup and drop-off locations of taxis) using geo-indistinguishability, a variant of differential privacy for location data. Finally, the demonstrator provides a web interface and web service to visualize the aggregated data to a data analyst.

We evaluate our demonstrator with regard to several criteria:

Performance is evaluated by measuring and comparing *latency* and *throughput* of collected messages between variants of the demonstrator that directly send messages from the clients to the server (no mix-net) or send them using the Panoramix mix network framework.

Usability is ensured by transparent anonymization (e.g., analytics do not have to be adapted between original and anonymized data) and a flexible Representational State Transfer (REST) Application Programming Interface (API) in combination with a web frontend for analytics.

Anonymity and Confidentiality are evaluated regarding communication metadata and the collected location data. First, without additional protection, communication metadata can reveal privacy-sensitive attributes such as the religion of a taxi driver. We measure how the integration of a mix network can reduce the success rate of such an attack. The mix-net furthermore provides confidentiality by using public key encryption for transferred messages. Second, learning that a user is part of a collection of location or mobility data can pose privacy risks since it allows inferring their location or habitual behaviour. We hence investigate the success of *membership inference* attacks on the collected data with and without geo-indistinguishability as protective measure.

Accuracy is evaluated with the analyst in mind who desires accurate analysis results despite the privacy-protective measures. We hence devise a metric based on the *earth mover's*

distance to quantify his loss in precision. Furthermore, we also employ the Jaccard index as practical utility measure that directly relates to potential business cases like ad placement or improving the services/availability oprovided by the taxi company.

Moreover, we align and validate the requirements formulated in the interim report (deliverable 6.1) on the finalized version of the demonstrator.

Contents

1. Introduction

The central objective of PANORAMIX is to design and develop a multipurpose infrastructure for privacy-preserving communications based on *mix networks* (mix-nets) and its integration into high-value applications that can be exploited by European businesses. Mix-nets protect not only the content of communications from third parties, but also obscure communication meta-data such as the exact identity of the senders or receivers of messages, through the use of cryptographic relays. What is more, they allow hiding from unauthorized third parties who is communicating with whom, when, where, and how often. Consequently, mix-nets are absolutely necessary for implementing strong privacy-preserving systems and protocols.

The goal of work package 6 is to demonstrate the use of mix-nets in one of these high-value applications: We want to demonstrate the use of mix-nets and differential privacy for privacy-aware data processing in the cloud. In this scenario, protecting the identity of data owners that submit data to the cloud is key to elicit truthful data (e.g. in surveys) and mitigate concerns against participation. As we will outline in higher detail, the selected use-case is involving data on taxi cruises. Thus, our objective is to support private gathering of data from the taxi sensors to the privacy-preserving compilation of real-time traffic maps or other smart city big data with about 1M-5M updates daily.

1.1 Scope and Purpose of Document

This deliverable reports on the final phase of WP6. It describes the final version of our demonstrator connected with the mix network and its evaluation with respect to privacy, utility, and performance. More specifically, it encompasses the following tasks from WP6:

Task 6.2 (System design and implementation) We will select the appropriate mechanism from WP3 for differential privacy and apply them either to the database client or database management system. We will use a highly scalable in-memory system in order to address the requirements of todays cloud infrastructure.

Task 6.3 (Integration with infrastructure) The database client and management system need to be connected via the mix network. We will adapt our interfaces to match those of WP4.

Task 6.4 (Validation & Testing) We will evaluate the performance, usability, anonymity, data confidentiality and accuracy of the demonstrator. This will help future commercially ready systems to tune their parameters.

According to these tasks, we have performed the following activities:

1. We have finalized our demonstrator and fully integrated the latest version of the Panoramix mix-net framework. This allows us to collect data from several taxi clients over the mix network instead of direct network connections, thus providing protection for communication metadata.

2. We have designed and performed several experiments to measure the benefits of both the mix network and the differential privacy technique for location data to improve privacy for the users of the system.

3. We have designed and performed experiments to evaluate the effects on utility of the proposed methods.

4. We have measured the performance of the fully-integrated simulation to see its impact on latency and throughput.

1.2 Relation to other Tasks and Deliverables

In the first and second year of the project, we conducted a requirements analysis (task 6.1) and devised the system design and implementation (task 6.2) of our demonstrator. Both were documented in the previously released interim report (deliverable 6.1).

Based on this initial design and implementation, we have completed our demonstrator and integrated it with the Panoramix mix-net framework (task 6.3) in the last phase of the project. Furthermore, we have validated and tested our demonstrator (task 6.4) regarding the requirements formulated in the requirements analysis. This final report (deliverable 6.2) summarizes the evaluation results and documents the work conducted in this final phase of the PANORAMIX project. Furthermore, it includes lessons learned and will provide guidance to future adopters of the system.

2. Use Case Description

As part of the PANORAMIX project, SAP is primarily working on the definition, implementation and validation of a use case which relates to a company transitioning its data and business operations into the cloud. An important driver for the cloud business is big data, where large amounts of information are aggregated and analyzed in order to extract value and provide new insights from the processed data.

The demand for data, however, is often faced with the data owners' reluctance to give out their data due to privacy reasons. Depending on the legislation, privacy regulations (such as the GDPR) might further prevent sensitive data from being shared or analyzed. A PANORAMIX goal is to provide European companies with tools to improve their business while protecting their own and their customers' privacy. To this end, we want to apply technical anonymization measures in order to convince data owners to share their data and to fulfil the necessary legal requirements. Anonymization could allow our customers to leverage client data that would previously have been unavailable for further analysis due to privacy concerns. This could give our customers better insights into their business or other activities.

2.1 Overall Goals

In work package 6, we want to demonstrate the use and advantages of the Panoramix mix network framework in a collaborative (SaaS) application. Such applications typically collect data (e.g. survey answers, sensor data from IoT devices) from a set of predefined (simulated) clients and aggregate those in a database. Due to the sensitivity of the data (e.g. health, religion, business secrets, etc.) it needs to be strongly protected. Still, we want to perform the typical big data type of aggregate analysis on them with reasonable accuracy. The overall objective of WP6 is to equip the database with the necessary mechanisms and connect it to the mix network.

We have identified several stakeholders at SAP whose use cases match this big data scenario where data from multiple sources is aggregated in a database in order to be analyzed. Among others, these include

- anomaly detection for enterprise systems (such as SAP Enterprise Threat Detection),
- evaluation of vehicle telematics data such as position for finding frequent routes, and
- evaluation of customer feedback in surveys or on social media.

In these applications, it is often required to incentivize data owners (i.e. customers) to share sensitive data with data analysts (i.e. service providers) for the sake of enabling new business models. For example, customers of a cloud service provider might be asked to provide feedback on the cloud service provider and may be reluctant to provide negative feedback, since they are dependent on the long-term business relationship.

Another example is benchmarking between multiple data owners, which would potentially provide beneficial performance insights, but could also be abused by competitors. Anonymity removes the link between contributed data and the corresponding data owner. Hence it encourages reporting, free from fear of retaliation. Still, in certain situations like an outstanding small or large company the data values themselves may reveal the data owner. Hence, we also want data confidentiality in order to protect them as well. Last but not least, we need performance to ensure scaling to the large volumes in a big data cloud scenario.

2.2 Identification of Frequent Taxi Pickup/Drop-Off Locations

In PANORAMIX, we will focus on the evaluation of position data from vehicles. The decision to focus on this use case was made as the underlying business logic is already mature and thus ensures a stable basis for research. Furthermore, stake holder interviews convinced us that this scenario best integrates scale, privacy and business aspects. The scenario is based on data collected from a fleet of taxis in real time. The data contains taxi trip information (e.g. pick-up locations and drop-off locations), which is then collected in a database for further analysis.

The data has the potential to realize efficiency gains through real time fleet management and planning, insights through periodic data mining, as well as data sharing with third parties such as advertising companies that wish to find the most frequented areas where they can place ads to attract maximal attention. However, it is also obvious that both passengers and taxi drivers face a high re-identification risk. Consequently, we want to protect both the identity of the taxis by hiding the source of the collected location (privacy on the network level), and the identity of the passengers by anonymizing the exact locations (privacy on the data level). Finally, a data analyst can analyze the collected and anonymized data, for example, to find the most frequent pickup/drop-off locations (hotspots). The insights on the identified hotspots could be used by the taxi company to improve their service by providing more taxis in (almost) real time or by an advertiser to place ads in those highly frequented areas without the previous inherent risk of violating customers' privacy and re-identification.

While the illustrated use case is built around taxi location data, it should serve as a generic example for a very general scenario that can provide value for many business cases: SAP previously offered a tool called "SAP Digital Consumer Insight", which basically provided data about pedestrian traffic around businesses (or any other location). It allows businesses to benefit from knowing about consumer demographics and behavior at a given location or point of interest. The insights provided allow users to run better marketing campaigns, improve advertising and services, scout locations and more. Moreover, from a technical point of view, the illustrated use case easily generalizes to many IoT scenarios where sensitive data from multiple clients is collected and aggregated in a central database.

3. Integration with Infrastructure

The PANORAMIX prototype for anonymization comprises a client (e.g., component for data owners) sending data through a mix-net (as provided by Panoramix) to a database (e.g., component administered by data curator). First, the client anonymizes geo-coordinates by applying geo-indistinguishability to each coordinate. This strong setup with privacy being enforced directly at the data source is defined as Local Differential Privacy. Second, the anonymized coordinates are send to the mix-net which encrypts the coordinates. Third, the mix-net delivers anonymized coordinates to the data analyst's database where coordinates are persisted. Analysts then perform queries over the data stored at the server. The message driven design realizes a real-time application.

A first version of the WP6 prototype integration was provided in Chapter 4 of the interim report D6.1 [WB17, p.19]. For the sake of brevity this deliverable will focus on providing information on changes in this WP6 prototype architecture. Main changes comprise additional considered location data from the Beijing Trajectory Project [ZFX+11] and a novel REST API at the server to allow dynamic queries for data analysts. We will furthermore present how initial requirements are addressed by the current architecture.

3.1 Integration with the Panoramix mix-net framework

Within the second half of PANORAMIX a new version of the mix-net was released. The change did not result in architectural changes but resulted in a speed-up from which requirements PERF-R1 and PERF-R2 (cf. Table 3.1) benefit.

The components have slightly changed due to an additional use case besides the Taxi data scenario introduced in D6.1. The additional use case considers coordinates generated by persons who participated in the Beijing Trajectory Project. The main difference lies in the nature of data that reflects multiple modes of transportation. The updated data flows are illustrated in Figure 3.3.

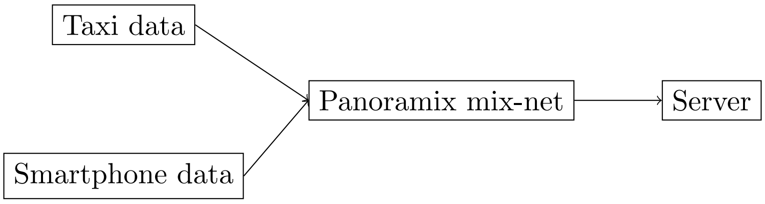
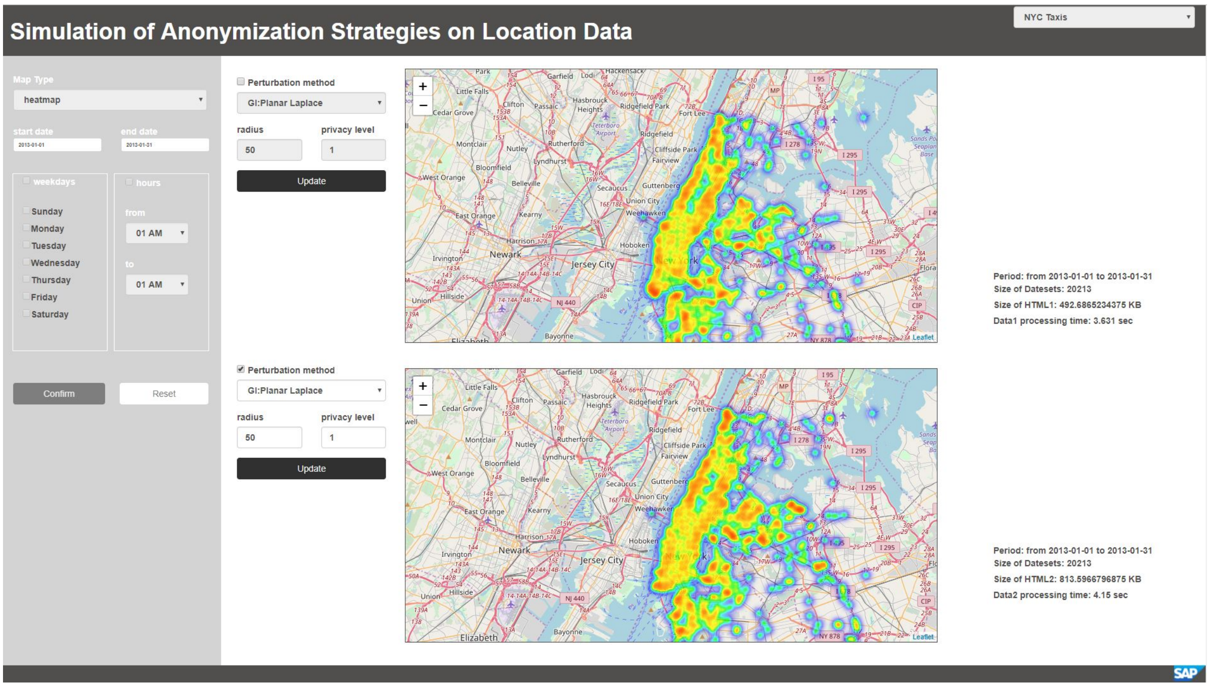


Figure 3.1: Simulation setup and dataflows with PANORAMIX mix-net.

3.2 Updates to the User Interface

A major change occurred at the server side where data can now be accessed by a wide range of users through a REST API. Furthermore, the prototype user interface was restructured and does now allow comparison of differential privacy mechanisms. In addition, shape files have been integrated to allow fine grained geographical queries. An illustration of the web user interface is provided in Figure 3.2.



Simulation of Anonymization Strategies on Location Data

NYC Taxis ▾

Map Type: heatmap ▾

start date: 2013-01-01 end date: 2013-01-31

□ weekdays
□ Sunday
□ Monday
□ Tuesday
□ Wednesday
□ Thursday
□ Friday
□ Saturday

□ hours
from: 01 AM ▾
to: 01 AM ▾

□ Perturbation method: GI:Planar Laplace ▾
radius: 50 privacy level: 1

Period: from 2013-01-01 to 2013-01-31
Size of Datesets: 20213
Size of HTML1: 492.6865234375 KB
Data1 processing time: 3.631 sec

☑ Perturbation method: GI:Planar Laplace ▾
radius: 50 privacy level: 1

Period: from 2013-01-01 to 2013-01-31
Size of Datesets: 20213
Size of HTML2: 813.5966796875 KB
Data2 processing time: 4.15 sec

Figure 3.2: Screenshot of the novel web user interface. Example comparison of two DP mechanisms.

The server now comprises a REST API to provide better support for novel projects as well as dynamic queries. In the following we introduce the Uniform Resource Identifiers (URI) and the functionality provided.

/server/panoramix Accepts POST messages and acts as server adapter. Once a message from clients is posted to this URL the bound function decrypts the received message and writes message content (e.g., a location) to the database.

/backend/homepage Produces the project webpage. This page is adapted dynamically by the backened query update component that is introduced below.

/backend/queryupdate Accepts POST and GET messages. This method interacts with JQuery.ajax() to generate "POST" requests that fulfil the choices (messages) from end-user like date or mechanism. Once the requests have been processed the JQuery.ajax() updates the maps in the project webpage without updating the webpage itself. We want to highlight that this components increases usefulness of the developed prototype for data analysts as dynamic comparison and selection (e.g., for certain weekdays and times) is possible.

3.3 Coverage of Requirements

In this section we illustrate how the final prototype covers the requirements identified in Chapter 3 of D6.1 [WB17, p.15]. Coverage is either achieved by certain design aspects and components. A detailed overview of implemented components is provided in Figure 3.3.

Qualitative requirements, such as ANA-R1, ANA-R2, CLI-R3 and PRI-R4 require additional detailed analysis due to the range of parameters that can be applied. This detailed analysis

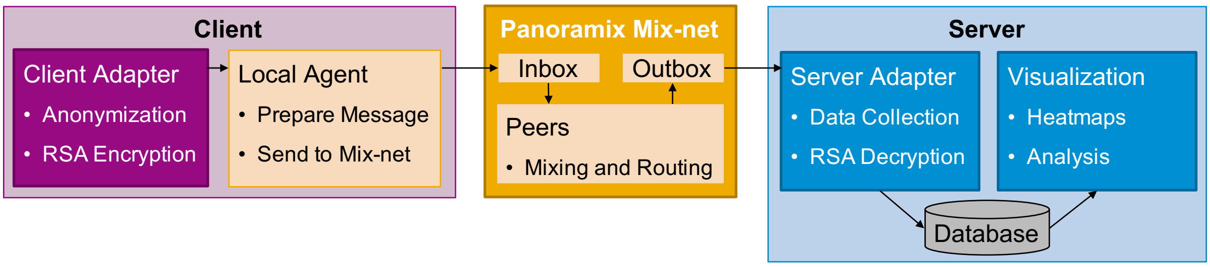


Figure 3.3: Overview of system components.

is provided in Chapter 4. For sake of completeness an overview of all requirements and the components by which they are addressed is presented in Table 3.1. All other requirements are covered by the four components listed below:

Client Adapter The client adapter was specified in D6.1. It covers differentially private anonymization of client data (e.g., locations), encryption of the anonymized data and submission to the Panoramix mix-net.

Panoramix mix-net The Panoramix mix-net is used as a black-box intermediary by our components.

Server Adapter The server adapter accepts and decrypts messages from the mix-net. Message content (e.g., anonymized locations) are stored in a database to be accessible for data analysts. The component was specified in D6.1.

Visualization Dynamic visualization is achieved through a website offering a REST API, s.t. content in the Server database can be dynamically filtered and aggregated for analysis. This component was extended in Section 3.2.

Table 3.1: Mapping of Requirements to Components.

Req. ID	Title	Component	Realization
CLI-R1	Submit Location	Client Adapter	HTTP push to Local Agent
CLI-R2	Network Privacy	Client Adapter, Panoramix mix-net	Routing data through mix-net
CLI-R3	Data Privacy	Client Adapter	Perturbing data using differential privacy
SRV-R1	Aggregate Data	Server Adapter	Message polling and insertion into database
ANA-R1	Visualize Hotspots	Visualization	Generation of heatmaps
ANA-R2	Real-Time Analysis	Server Adapter, Panoramix mix-net	Storing data in an in-memory database, Efficient mix-net implementation (deferred to mix-net developers)
PRI-R1	Network Privacy	Panoramix mix-net	Routing data through mix-net
PRI-R2	Collusion Resistance	Panoramix mix-net	Using three or more mix-net servers
PRI-R3	End-to-End Encryption	Client Adapter, Server Adapter	Public-key encryption (RSA) between client and server

PRI-R4	Data Privacy	Client Adapter	Perturbing data using differential privacy
PRI-R5	Untrusted Server	Client Adapter, Panoramix mix-net	Perturbation at the data source (local differential privacy), Routing data through mix-net
PERF-R1	Mix-Net Throughput	Panoramix mix-net	Efficient mix-net implementation (deferred to mix-net developers)
PERF-R2	Mix-Net Latency	Panoramix mix-net	Efficient mix-net implementation (deferred to mix-net developers)
PERF-R3	Database Efficiency	Server Adapter	Storing data in an in-memory database
PERF-R4	Performance Evaluation	Client Adapter, Server Adapter	Inclusion of timestamps in sent messages (only for evaluation)

4. Validation and Testing

In this section, we describe additional experiments we performed to evaluate the effects of our privacy-preserving technologies on privacy and utility. Furthermore, we measure the performance impact of integrating the mix-net in our demonstrator.

4.1 Utility

In our scenario, protecting the privacy of the involved users (i.e. taxi drivers and passengers) is a main concern. However, we also want to make sure that the collected data remains useful for further processing and analysis. In particular, we want that our collected taxi data can still be used by an analyst to provide valuable insights for customers, the public, or other third parties. Therefore, to measure the effects of the privacy-protective measures, we describe and evaluate utility of the obfuscated data in terms of utility for ad placement. This also supports requirements ANA-R1 (visualization of hotspots) by giving quantitative measures for the accuracy of the heatmaps and hotspots.

Utility of ad placement Consider an advertising company that needs to know the best location to place their ads. This could correspond to the most frequented locations ("hot spots") where people take taxi rides. If we divide the area of interest into several small cells, an analyst could take the collected location data and compute a heatmap corresponding to the number of taxi pickups and drop-offs in each cell, and then share the heatmap or the top hot spots with the advertising company. The advertisers could then place their ads in the most frequented hotspots in order to maximize the number of consumers.

Accuracy measures To examine the usefulness of the anonymized data, we could naïvely inspect the differences in the heatmaps (cf. figure 4.1) or histograms (cf. figure 4.2) between original and anonymized data. However, while visual inspection can serve as a rough indicator of the effects of the anonymization, it is a rather vague measure. To make this more tangible, we will use the earth mover's distance (EMD) and Jaccard index (defined below) as *quantitative* accuracy measures for the deviations in the heatmaps and the number of the top $k\%$ most frequented cells that are retained after obfuscating the location data, respectively.

4.1.1 Experiment Setup

This section describes the common setup and preparation steps we perform before conducting the experiments with the earth mover's distance and Jaccard index to measure the effect of data obfuscation on utility in the ad placement scenario.

Data preparation Because of the huge scale of the New York Taxi dataset, we only choose trips that happened between April 1st and 5th (Monday to Friday), 7 to 10 AM (morning rush hour), which amount to around one million points. To obfuscate the collected location data, we

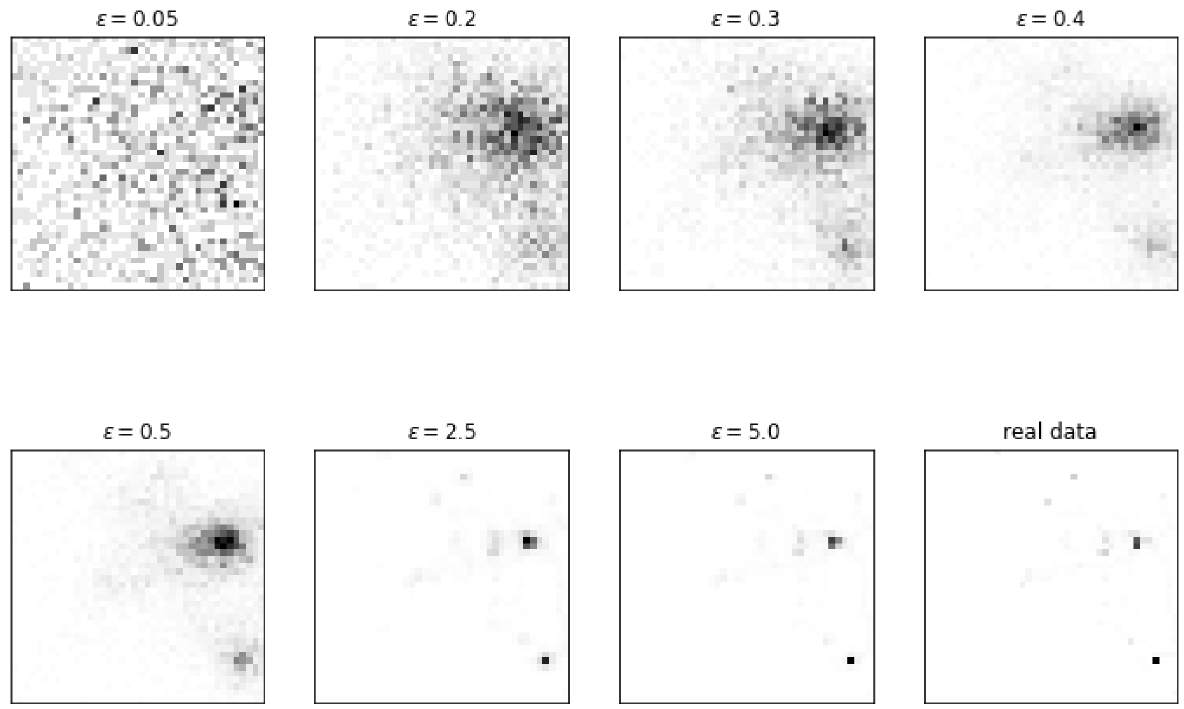


Figure 4.1: Heatmaps (grid resolution $N = 40$) from one user in the New York Taxi dataset obfuscated with different ϵ and original data.

apply the planar Laplace mechanism with protection radius $r = 200$m and privacy parameter ϵ ranging from 0.05 to 5.

Heatmap generation Aiming at comparing the most frequented locations between the original and obfuscated data, we adopt a 2D grid that partitions the observed region R into $N \times N$ cells, where N a is a constant called *resolution*. We choose the length of a cell ranging from 1235m to 200m, which corresponds to a resolution N of about 10 to 60. Based on the original, unperturbed data, we denote the frequency score (number of pickups/drop-offs) of each cell at position (i, j) in this region as $x_{i,j}$. Thus the heatmap for the original data is given as the set $X = \{x_{i,j} : 1 \leq i, j \leq N\}$ of all the cell frequencies $x_{i,j}$ in the grid. Similarly, the heatmap for the obfuscated data is given as the set $Y(\epsilon) = \{y_{i,j}(\epsilon) : 1 \leq i, j \leq N\}$, where ϵ is the privacy parameter controlling the noise induced by the planar Laplace mechanism. If ϵ is fixed we can omit it and simply write $y_{i,j}$ and Y for the obfuscated cell frequencies and corresponding heatmap, respectively.

Experiment conduction Based on the heatmaps for the original and perturbed data, we conduct the following experiments to measure utility based on the earth mover's distance and Jaccard index.

4.1.2 The Earth Mover's Distance

The earth mover's distance (EMD) is a measure of the distance between two distributions X and Y over a domain R. Intuitively, if the distributions are interpreted as two different set of piles, the EMD is the minimum cost of moving one set of piles into the other. In our case, we assume the domain R to be discrete, thus the EMD can be computed by solving a classic *transportation problem* [Hit41].

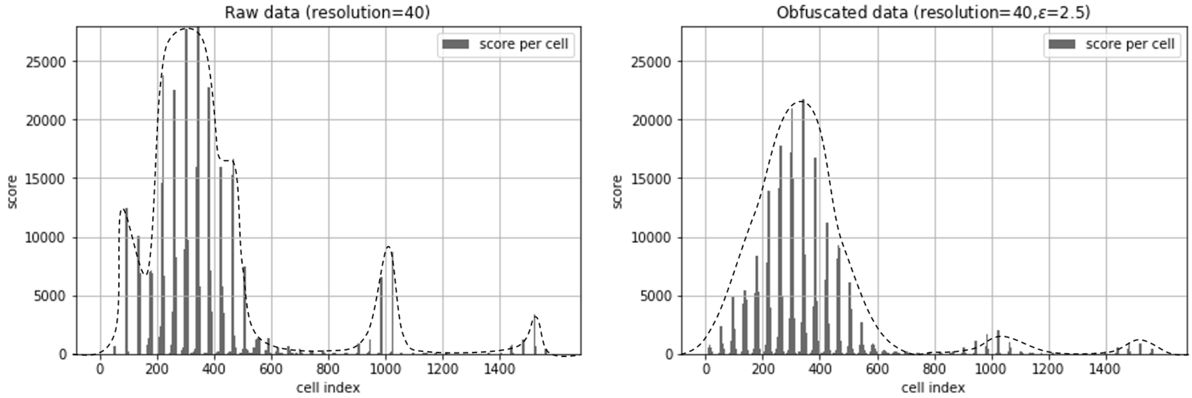


Figure 4.2: Flattened histogram from New York Taxi dataset.

According to [RTG98], we give our definition as follows:

Definition 4.1 (Earth mover's distance)**.** Given two one-dimensional arrays $\mathcal{X} = [x_1, x_2, ..., x_m]$ and $\mathcal{Y} = [y_1, y_2, ..., y_n]$ representing the two distributions and ground distance $D = [d_{ij}]$, we find the optimal flow f_{ij} between x_i, y_j through the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij} \\ \text{subject to} \quad & f_{ij} \geq 0 \quad \forall\, 1 \leq i \leq m,\ 1 \leq j \leq n, \\ & \sum_{j=1}^{n} f_{ij} \leq x_i, \quad \sum_{i=1}^{m} f_{ij} \leq y_j, \\ & \sum_{i,j} f_{ij} = \min\left\{\sum_i x_i, \sum_j y_j\right\}. \end{aligned}$$

The *earth mover's distance* between X and Y is then defined as

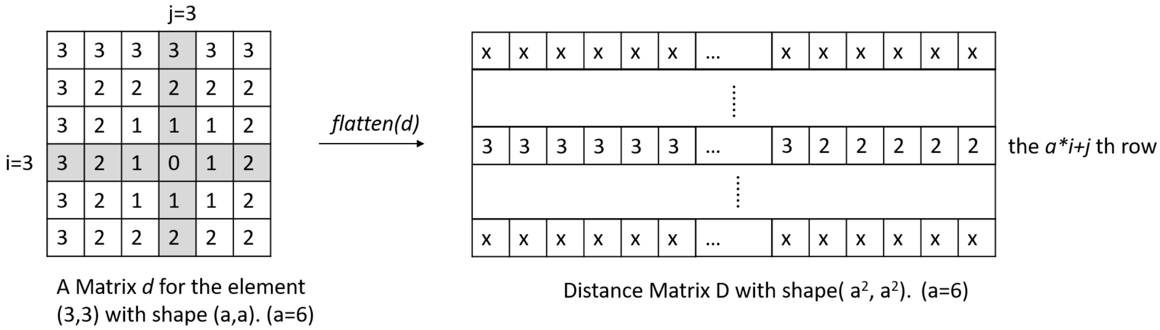
$$\text{EMD}(X,Y) := \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}.$$

Pele and Werman [PW08] proved that EMD is a metric for any two one-dimensional arrays if the ground distance D is a metric. It can be easily solved for using linear optimization methods.

EMD ground distance matrix Before we can compute the earth mover's distance of the heatmaps X and Y, we still need to specify a distance matrix D that defines a distance/cost between all pairs of cells (i, j). Since there are N^2 cells in total, D will have shape $N^2 \times N^2$. Intuitively, we define the ground distance matrix D for each element (i, j) according to the Manhattan distance so that neighboring cells have distance 1, as illustrated in figure 4.3. Finally, we have everything we need to compute the earth mover's distance of the original and perturbed heatmaps X and Y, based on the ground distance matrix D. In our implementation, we use the PyEMD Python package to perform the computation.

Experiment results

As can be seen from figure 4.4, all EMDs start with high values and plunge to $\epsilon = 0.5$ (decreasing deviation/increasing utility) and then fall continually but not as rapidly as before. Note that


j=3
i=3
flatten(d)
the a*i+j th row
A Matrix d for the element (3,3) with shape (a,a). (a=6)
Distance Matrix D with shape(a², a²). (a=6)

3	3	3	3	3	3
3	2	2	2	2	2
3	2	1	1	1	2
3	2	1	0	1	2
3	2	1	1	1	2
3	2	2	2	2	2

x	x	x	x	x	x	...	x	x	x	x	x	x
⋮												
3	3	3	3	3	3	...	3	2	2	2	2	2
⋮												
x	x	x	x	x	x	...	x	x	x	x	x	x

Figure 4.3: Construction of the distance matrix D

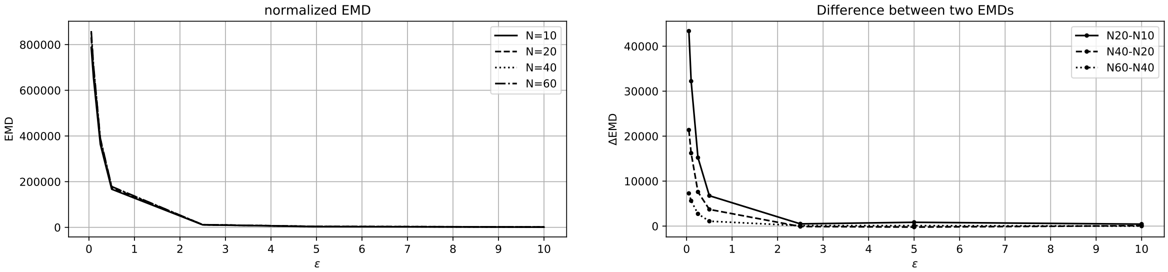


Figure 4.4: EMD for different resolutions N and ϵ

differences in cell size are not very pronounced, and that it can only give a rough quantitative guidance of how much the heatmaps deviate between original and obfuscated data.

To get an idea how much the obfuscation affects a specific kind of utility, it is therefore better to measure the impact directly. We do this in the following experiments with the Jaccard index, which shows the same tendencies as we will see e.g. in figure 4.5: All lines have an upward trend (improved utility) with the increase of ϵ. We emphasize that with $\epsilon \geq 0.5$ we obtain acceptable utility in our case.

4.1.3 The Jaccard Index

The Jaccard index [Jac01, Kos16] is a classical similarity measure on two finite sets. It is formally defined as follows:

Definition 4.2 (Jaccard index)**.** Given two sets A and B, their *Jaccard index* is

$$\mathcal{J}(A, B) := \frac{|A \cap B|}{|A \cup B|},$$

where $\mathcal{J}(\emptyset, \emptyset)$=1.

Obviously, the larger $\mathcal{J}$ is, the more similar the two sets are. When determining the most frequented areas or cells is the desired goal of the analyst, for instance to provide improved services or for marketing purposes (ad placement), we can therefore use the Jaccard index to rate the agreement of top areas between original and obfuscated data.

Starting with the heatmaps of the original and perturbed data $X = \{x_{i,j} : 1 \leq i, j \leq N\}$ and $Y(\epsilon) = \{y_{i,j}(\epsilon) : 1 \leq i, j \leq N\}$, we rank the cells decreasingly by their frequency scores $x_{i,j}$ and $y_{i,j}$ to find the highest density cells in the considered area. For $0 < k \leq 100$, we find the top $k\%$ cells with the highest scores for the original and perturbed data, denoted by $T_k(X)$ and $T_k(Y)$. We then compute the Jaccard index

$$\mathcal{J}(T_k(X), T_k(Y))$$

to see how many of the top $k\%$ cells are retained in the statistic between the original and perturbed data, where a higher Jaccard index indicates a better preservation of the utility e.g. for the advertiser with whom the heatmap or top $k\%$ data is shared.

Experiment results

We examine the Jaccard index of the top $k\%$ cells for different values of N and ϵ, respectively. From figure 4.5, we can see that all Jaccard index values increase with ϵ. This is expected since larger ϵ provides more accuracy, however at the cost of less privacy protection.

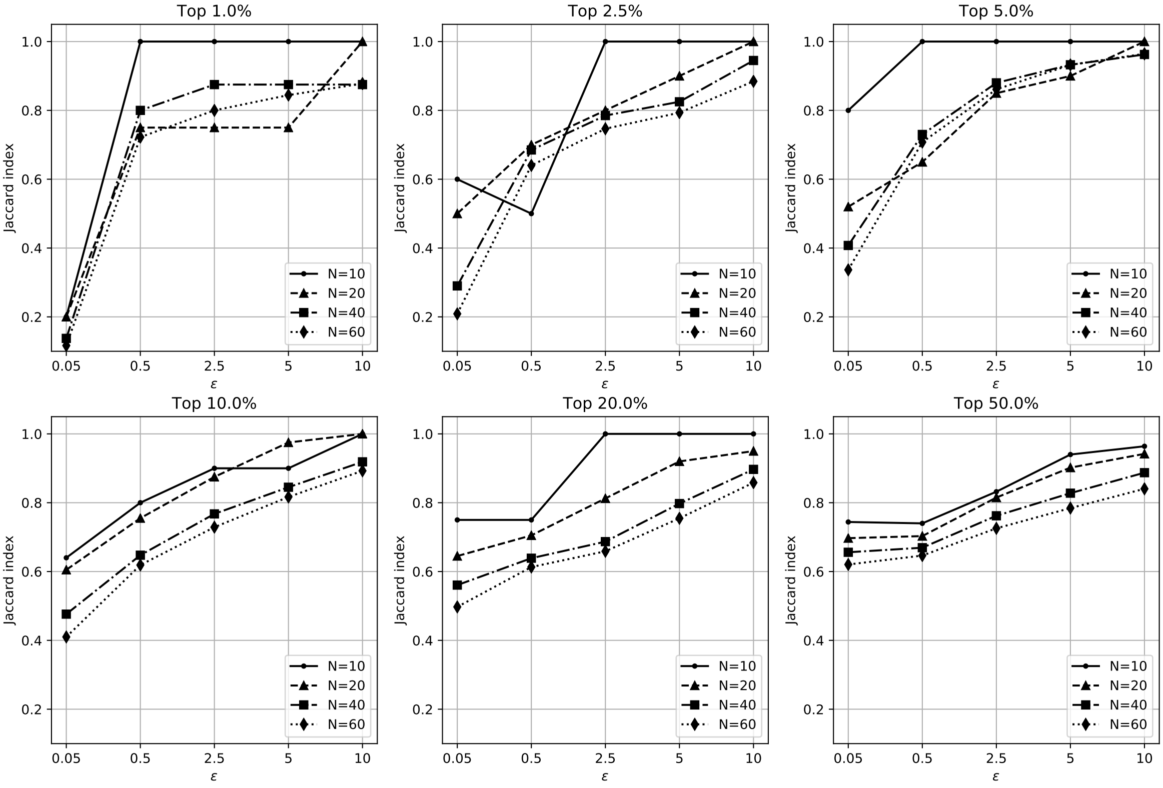


Figure 4.5: Jaccard index by different percentages of top cells

Regarding the resolution parameter N, we observe a trend that larger cell sizes (smaller N) typically provide better utility, with some exceptions due to fluctuations in the obfuscated data. This is because the perturbations caused by the Planar Laplace (PL) mechanism cause fewer points to move to different cells as the cells are larger. We can observe this effect more clearly in figure 4.6.

Similarly, when considering a larger percentage or number of most frequented cells, the likelihood that a cell that is part of the top cells based on the original data also is a top cell with the obfuscated data generally increases. Even very small ϵ can guarantee a Jaccard index over 0.6 for all resolutions N if we choose top 50% of the cells.

Taken together, this case study implies that getting stronger privacy protection with smaller ϵ using the Planar Laplace mechanism results in some loss of utility. However, in this concrete application, with an appropriate choice of $\epsilon \geq 0.5$, the Planar Laplace mechanism can still product good enough utility of over 70% if the cell size is not too small.

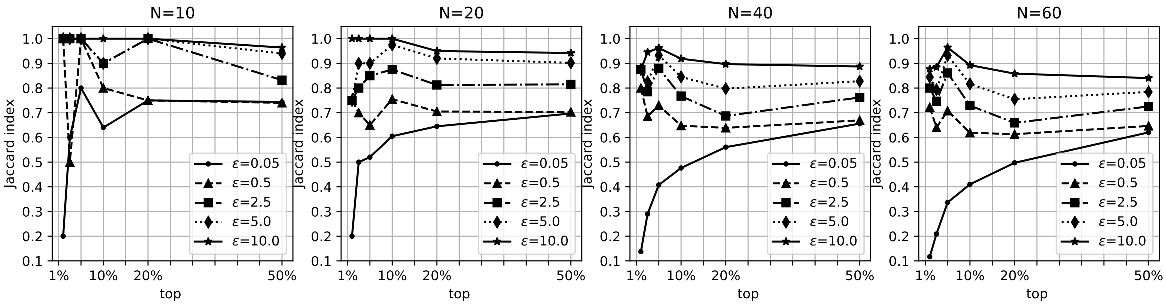


Figure 4.6: Jaccard index by different resolutions N

4.2 Attacks

Releasing only aggregate statistics on a dataset is insufficient to protect individuals in the dataset from re-identification attacks. Thus we use Differential Privacy to prevent against singling out attacks (e.g., such as formulated by Dwork and Roth [DR13, p.8]) based on aggregates. However, the protection of Differential Privacy foremost depends on the parameter ϵ and is relative, i.e., only holds in the context of a function $f(\cdot)$ (e.g., count query). This indicates the need to provide users with other metrics to interpret their actual protection. Furthermore, to judge how well the qualitative requirements CLI-R3 and PRI-R4 are covered (cf.Table 3.1) we need quantifiable and understandable metrics.

We selected two attacks for privacy evaluation. First, a membership inference attack that strives for identifying individuals within the dataset with high probability. Second, a religion inference attack that strives for inferring a sensitive attribute (religion) from the trip data of individual taxis.

4.2.1 Membership Inference

In the scope of PANORAMIX we apply a membership inference attack (MIA) model based on the experiment of Pyrgelius et al. [PTDC17] to provide an additional, tangible interpretation besides the abstract DP guarantee. In this specific membership inference attack an adversary, *Bob*, possesses some prior knowledge p about a target user $u^* \in U$ (e.g., *Alice*), where U represents the set of all users. The prior knowledge p expresses Bob's knowledge about locations visited by Alice within a certain period of time $\mathcal{T}^*$

We encode Bob's prior knowledge as a triplet (u^*, p, t), where $t \in \mathcal{T}^*$. Conceptually, Bob desires to infer whether Alice's data is contained within the released aggregate statistic using his prior knowledge p. Technically the membership inference attack itself is realized by a machine learning classification model. The training dataset is generated from p, and the test dataset is the set of obtained aggregates. The attack is structured as follows.

Generating in/out user-groups To generate user groups for classification U is split into two subsets, each of size m, containing user histograms (i.e., their distribution).

1. Randomly sample m users' histograms and arrange in one user-group G^{out}.

2. Randomly sample $m - 1$ users' histograms and u^*'s histogram as one user-group G^{in}.

Feature Extraction Two datasets, D^{out} and D^{in}, are created for training as well as testing of the classifier. The following process is executed n times based on the size of set.

1. Extract features *min*, *max*, *median*, *average*, *standard deviation* per grid cell in G^{out}, G^{in}. Consequently, we obtain $5N^2$ features per user-group arranged in feature arrays per entry F^{out} and F^{in}.

2. Append the feature array and the corresponding label, either 0 for excluding u^* or 1 for including u^*.

Binary classification The actual membership inference attack is performed by a binary logistics regression (LR) classifier which after training predicts $\{in, out\}$ for test records.

1. To train the classifier we create a balanced training dataset of size n from the two subsets $D^{out} = \{(F_1^{out}, 0), \ldots, (F_n^{out}, 0)\}$ and $D^{in} = \{(F_1^{in}, 1), \ldots, (F_n^{in}, 1)\}$. Thus, $|D^{out}| = |D^{in}|$.

2. To test the classifier we likewise build a balanced test dataset on unseen user-groups. Testing yields the accuracy rate x on the test dataset. We define the performance score of this model as P:

$$P = \begin{cases} 0 & \text{x}\leq 0.5 \\ \text{(x-0.5)/0.5} & \text{x}>0.5 \end{cases}$$

   since x=0.5 is the baseline of this binary LR prediction task.

Forethought

Pyrgelis et al. [PTDC17] show that membership inference is successful when the adversary possesses prior knowledge in the form of a small subset of user events. Thus, we assume that the membership inference model introduced before (cf. Section 4.2.1) works well when applied to a real dataset D_{real}. Furthermore, we run the same model repeatedly for different perturbed datasets D_ϵ, produced by applying the PL mechanism with fixed $r = 0.2$ km and varying ϵ (cf. figure 4.7) on D_{real}. Finally we compare the results between D_{real} and D_ϵ.

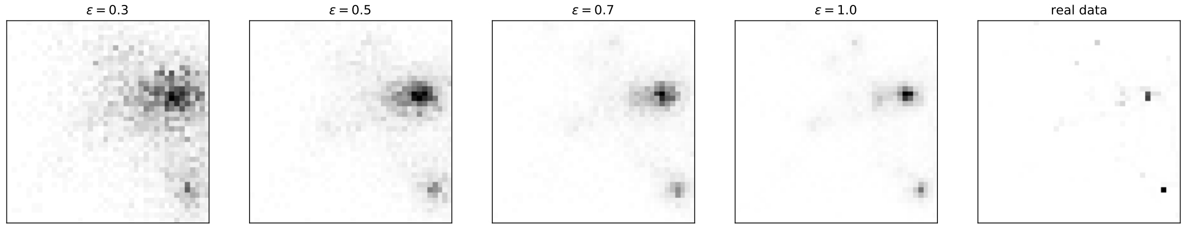


Figure 4.7: Events from one user during a period of time shown in $40 \cdot 40$ grid with different ϵ

Setting

We consider the sets D_{real} and D_ϵ for all events in the period T spanning from May 2008 to August 2009. During this period the set U comprised 104 active users. We divide D_{real} into three subset D_{real}^{prior}, D_{real}^{target} and D_{real}^{unseen} $((D_{real}^{prior} \cup D_{real}^{target})\complement = D_{real}^{unseen})$. Furthermore, we assume that the adversary knows the locations of 25 (out of 104) users in the period $\mathcal{T}$, as well as his target u^*. In other words, we simplify our prior knowledge setting where the inference and the observation period completely coincide (i.e., both are $\mathcal{T}$). We then generate a balanced training dataset by randomly sampling $n = 400$ unique aggregation groups (i.e., $|G^{out}| = 200$, $|G^{in}| = 200$) from D_{real}^{prior}.

Attack on D_{real} First, we create a balanced testing dataset by randomly sampling 100 unique user groups from D_{real}^{unseen}. Second, we extract features for each aggregation group and label it as mentioned at the beginning of Section 4.2.1. Third, we run experiments with $m = \{5, 10, 15, 20\}$ in order to evaluate the effect of aggregation group size.

Attack on obfuscated D_ϵ In respect to D_ϵ we examine whether our model still yields a good performance, compared to D_{real}, score under different for varying ϵ. The results are depicted in figure 4.8 and figure 4.9. Due to the introduced perturbation we generate D_ϵ 10 times per ϵ and calculate the average performance score after testing each generated D_ϵ.

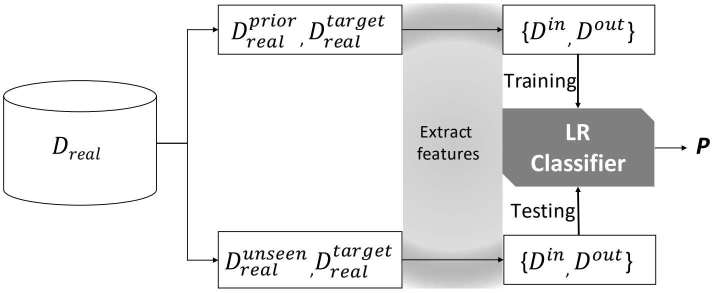


Figure 4.8: MIA model diagram (real dataset)

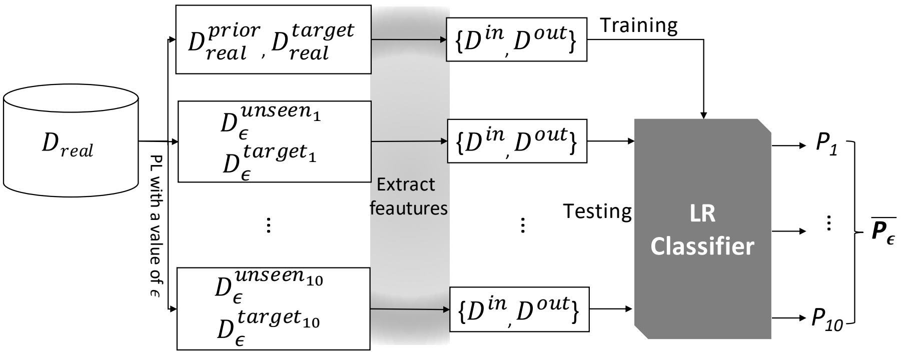


Figure 4.9: MIA model diagram (dataset with PL mechanism)

Results

Figure 4.10 states the results after having performed the attack on D_{real} for each target $u^* \in U$. It can be seen that the model achieves high performance scores $P \geq 0.8$ for 37 users. We select these users with $P \geq 0.8$ as the targets for the attack on the dataset D_ϵ perturbed with PL. The results presented in figure 4.11 and figure 4.12 represent the average P for these target users.

As we can see in figure 4.11, the score P of D_{real} decreases when the aggregation group size m increases. In contrast, when $\epsilon \leq 0.7$ the performance scores are not affected by m. More importantly, with $\epsilon \leq 0.7$ the PL mechanism can effectively reduce the success rate of MIA and thus mitigate the attack.

4.2.2 Religion Inference Attack

When membership in a dataset is known an adversary might have specific interest in inferring sensitive information about an individual in the database. One such example is the case of inferring religious affiliation based on metadata, which we refer to as Religion attack. The attack specifically desires to identify Muslims in the database with high likelihood by exploiting the publicly available background knowledge on Salah times (prayer times). The religion attack

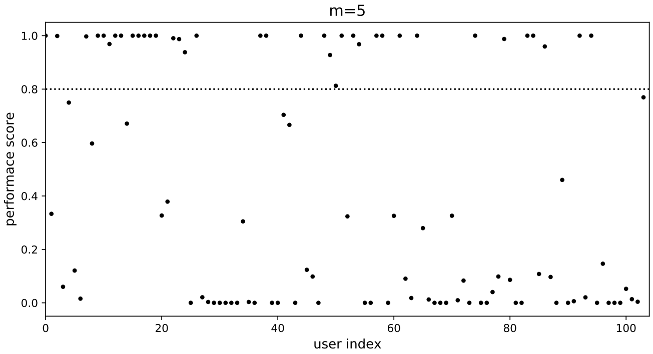


Figure 4.10: Original MI performance scores of 104 users

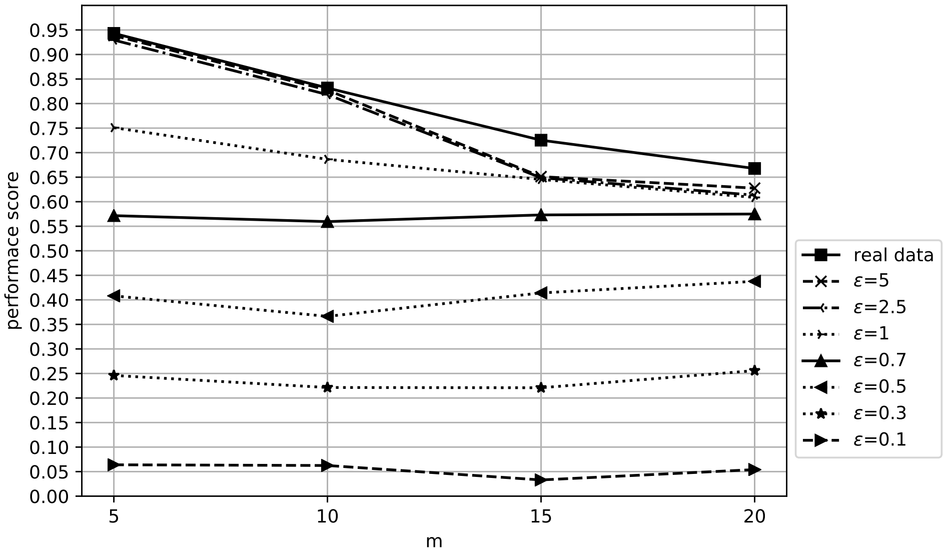


Figure 4.11: MI performance scores for varying values of ϵ and aggregation group sizes m

is possible independently of the differentially private location data by observing that there are no reported locations within a given timeframe / at given timestamps (e.g. during the midday prayer time). From a privacy perspective it is clear that such an attack can reveal personally sensitive information and can be used for discrimination; hence it should be prevented.

Within PANORAMIX, mix networks are best suited for protection against such attacks by (delayed) bulk message delivery that hides the exact times of the drivers' activity.

Attack Model

In our taxi driver scenario, due to the real-time requirement, the timestamp when a taxi reports an event discloses the actual time either the driver picks up or drops off a passenger. A malicious, honest-but-curious server operator can thus learn the periods when a taxi driver is *busy* transporting passengers. Combining this information about individual taxi drivers with publicly known Islamic prayer times (for NYC 2013) [Pra], he can figure out whether a taxi driver likely is a Muslim or not.

Time blocks It is generally agreed that each prayer should take about 5 to 15 minutes. We hence divide each day into 10 minute blocks and, for each driver and day, mark each block as

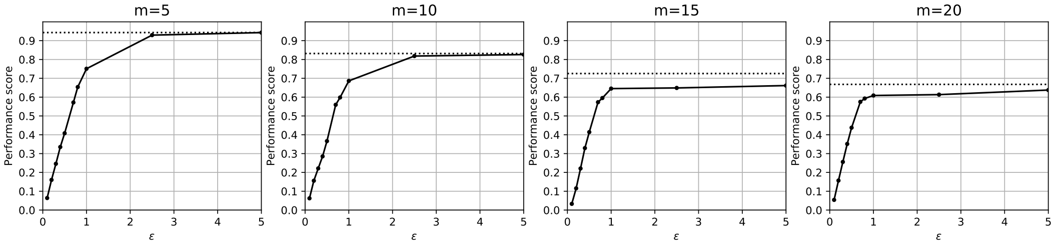


Figure 4.12: MI performance scores under different m by increasing ϵ

busy or *idle* depending on whether it lies between a pair of pickup and drop-off events or not. For each day this yields a time block vector with $24 \cdot 6 = 144$ blocks. Figure 4.13 and figure 4.14 visualize the activities from two taxi drivers in January 2013, where each row represents one day. Dark areas mark the time blocks when the taxi was idle (not taking passengers). On the contrary, the gray blocks signify that the driver was busy driving passengers. The five white lines refer to the five daily prayer times, Fajr, Dhuhr, Asr, Maghrib, and Isha, which slightly change every day.

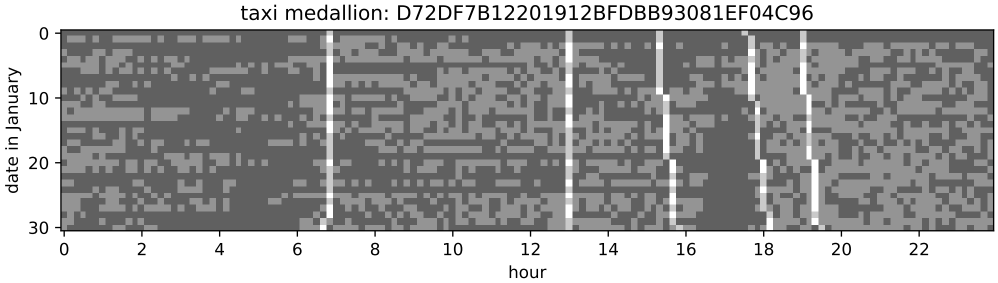


Figure 4.13: Activities from a randomly chosen taxi

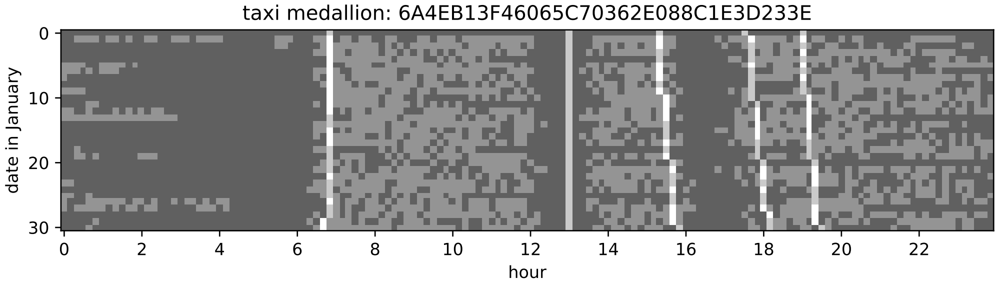


Figure 4.14: Activities from a "possible Muslim"

It is apparent in these two figures that the driver's patterns are different. For the second taxi, we see that the driver's idle times align very well with prayer times so he is much more likely to be a devout Muslim than the first one. Therefore, if the adversary knows the precise idle and busy periods of a taxi driver, he can identify (with high possibility) whether the driver a Muslim or not.

For a driver d, we define a binary *time block matrix* $M = M(d)$ where each row stands for a day with $24 \cdot 6$ blocks. Each entry $M_{i,j}$ indicates whether the driver was busy during the j-th time block on day i, i.e.

$$M_{i,j} = \begin{cases} 1 & \text{if the driver had passengers (was busy) in the corresponding time block,} \\ 0 & \text{if the driver had no passenger (was free) in the corresponding time block.} \end{cases}$$

Similarly, we represent the official Islamic prayer times (for New York City in January 2013) as binary time block matrix T, where each entry indicates whether the corresponding time block falls within a prayer time or not, i.e.

$$T_{i,j} = \begin{cases} 1 & \text{if this timeblock is prayer time,} \\ 0 & \text{otherwise.} \end{cases}$$

Subsequently, we compare T and M to determine a daily *busy sum* to decide about the driver's religion (cf. figure 4.15). The busy sum for a day i can simply be computed as scalar product between the vectors in row i of T and M, i.e. $\langle T_i, M_i \rangle$.

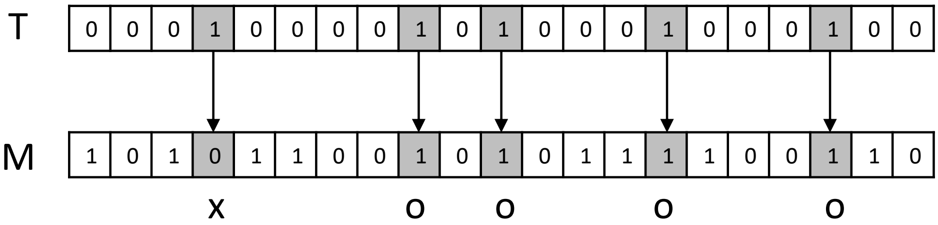


Figure 4.15: Comparing prayer and busy times

Busy scores Basend on the activities of the 1000 drivers in January 2013, the *average* daily busy sums for the five prayer times are 8.82, 14.65, 14.49, 14.97, and 17.6, respectively. This implies that different prayer times have a different weight in the decision whether a driver is a Muslim. Aiming to quantify the possibility of identifying a Muslim, we determine the sum of busy score for each driver d as

$$s(d) = \sum_{k=1}^{5} c_k \cdot \langle T, M(d) \rangle$$

where $\langle \cdot, \cdot \rangle$ is the scalar product and $c_k = \frac{\sigma_k}{\mu_k}$ is the *coefficient of variation* of the k-th prayer time. The coefficient of variation (cf. [Eve02]) is commonly used in setting weight and is defined here as the ratio of the standard deviation σ_k to the mean μ_k for the k-th prayer time over all 1000 taxi drivers. Obviously a lower busy score indicates a higher probability to be a Muslim.

Experiment Setup

To evaluate the religion inference attack, we choose the top 1000 taxi drivers that have more than 1000 events in January 2013. We thus obtain data consisting of roughly 2,600,000 events, corresponding to an average of about 1 event per second for that month. Using these preprocessed events and their original timestamps, we can calculate the original busy score for each driver as described above, and then arrange the 1000 drivers by busy score as depicted in figure 4.16 to get a "possible Muslim" ranking that we interpret as ground truth.

Mix networks as countermeasure We know that using a mix network produces additional *latency overhead* which depends on two factors: Firstly, the mixing operations of the mix-net itself (cryptographic operations, relaying messages between peers, etc.) are computationally intensive and hence cause delays. Secondly, there is a waiting time until enough messages have arrived at a mixing node before it will start its mixing operations. The second factor is controlled

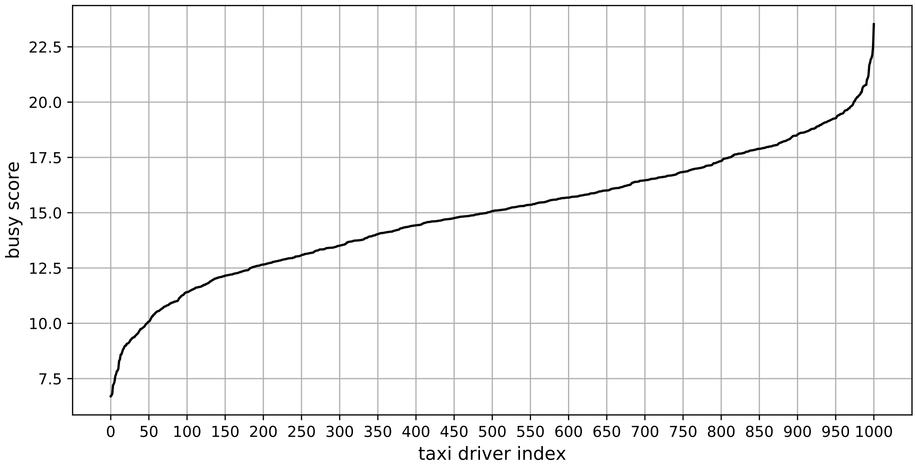


Figure 4.16: Busy scores for 1000 taxis (sorted)

by a parameter called *batch size* which specifies the minimum number of messages before mixing starts. In our evaluation, we will evaluate the effect of the batch size on distinguishing a Muslim.

We simulate sending the pickup and drop-off events of those drivers over a mix network consisting of three peers with different batch sizes, and record the delayed timestamps upon receiving each event.

Results

For the evaluation, we calculate the busy scores based on the delayed timestamps and compare the ranking with the ground truth we obtained with the original timestamps. We make an assumption that $k\%$ of the drivers are Muslims, where we chose k as 5%, 10%, 15%, or 20% representing 50, 100, 150, or 200 of the 1000 total drivers. For example, in figure 4.16 we can see that the 200 taxi drivers (top 20% from 1000) with the lowest busy scores have a busy score below 13. By utilizing the Jaccard index to compare between the top k% drivers with lowest original and delayed busy scores, we can finally evaluate how accurate the results are when using a mix network.

The results of the experiments with the Jaccard index are depicted in figure 4.17: For all tested values of k, the proportion of drivers correctly (based on our ground truth) identified as Muslim drops with increasing number of messages in a batch. Furthermore, we can see that the drop is quite substantial for the top 5% even when smaller batch sizes are used. This shows that for protecting a small percentage of the population, a mix network with even a small batch size can be sufficient. On the other hand, by setting a larger batch size we can manage to achieve good privacy protection which affects the attacker's performance by over 30% also for larger proportions of the drivers, at the cost of getting somewhat larger delays in the message transportation. However, if we assume at least 1,000,000 messages per day in a realistic setting, even a batch size of 300 would only produce an extra delay of roughly 30 seconds on average.

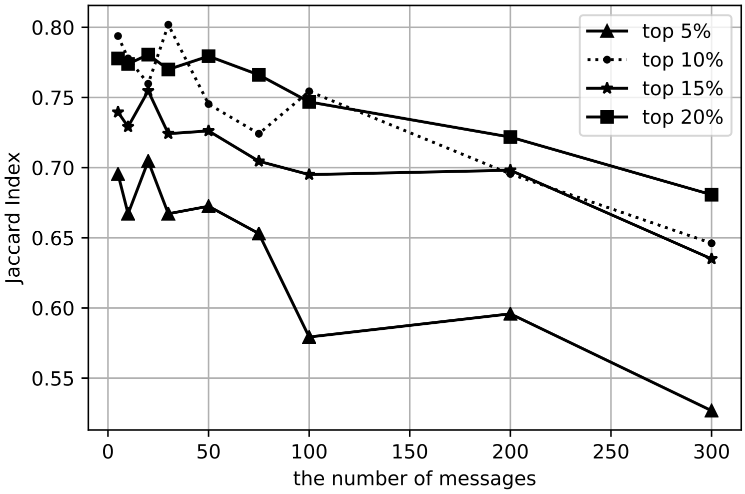


Figure 4.17: Jaccard index for different batch sizes (number of messages) assuming a top $k\%$ Muslim population

4.3 Performance

In this section, we describe experiments measuring the performance of our system. The goal is to support requirements ANA-R2 (realtime analysis), PERF-R1 (mix-net throughput), as well as PERF-R2 (mix-net latency) with quantitative performance figures that confirm the feasibility of using both privacy-enhancing technologies, mix networks and differential privacy, in a realistic data collection and statistics setting. Furthermore, as stated in our grant agreement *"our objective is to support private gathering to compile real time traffic maps or other smart city big data about 1M-5M updates daily"*, which we also want to verify with the experiments that follow.

We evaluate latency and throughput for several configurations of the mix network and the simulation. As major parameters, we examine the effects of the *number of peers* in the mix-net, its *batch size* (minimum number of messages collected at a node before mixing starts), and the *input rate* describing how many messages per second are sent into the mix-net.

4.3.1 Preliminary Experiments

We perform initial experiments directly on a "contemporary" developer laptop powered by an Intel Core i5 8350U CPU with 4 physical cores (8 threads) and 16GB of RAM. We use Panoramix to setup a mix-net with batch size 100 consisting of three peers, and vary the input rate between 8 and 24 messages per second. In figure 4.18, we can see that the number of messages fed into the mix-net has a strong effect on performance where both throughput and latency grow with increasing input rate. Furthermore, we make the interesting observation that when sending more than 20 messages per second, latency and throughput suddenly grow much faster than with lower input rates.

To investigate this further, we compare feeding 12 and 24 messages per second into the mix-net in figure 4.19. With an input rate of 12, the network is able to provide a steady message throughput of 12.63 messages per second and a latency of 17.62 seconds on average (which is the minimum latency over the evaluated input rates). Now if we double the rate to 24 messages per second, we observe a rising throughput but also increasing latency over time. However, notice that the network breaks down and fails early after running for around 300 seconds. Therefore,

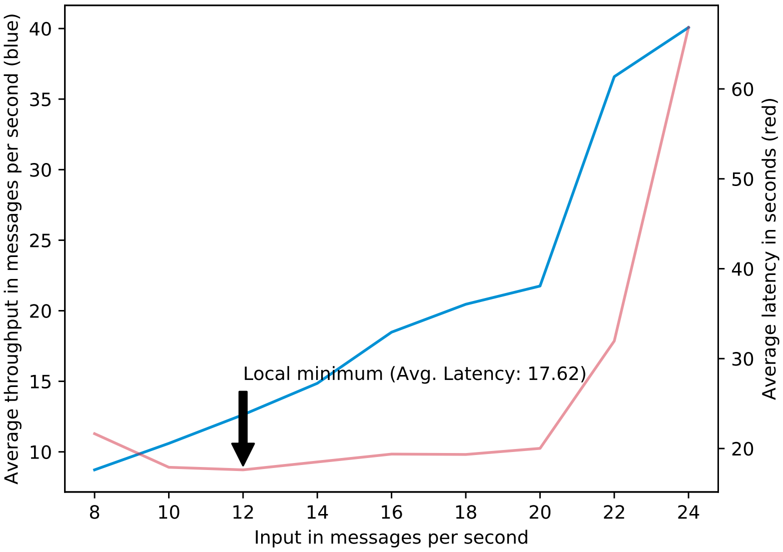


Figure 4.18: Throughput and latency of a 3-peer mix-net with batch size 100 (preliminary result on developer laptop).

to allow continuous operation of the mix-net, we observe that the input rate must be kept below a certain threshold that depends on the underlying hardware (i.e. 20 messages per second on the developer laptop) in order to prevent flooding the network and avoid service outages.

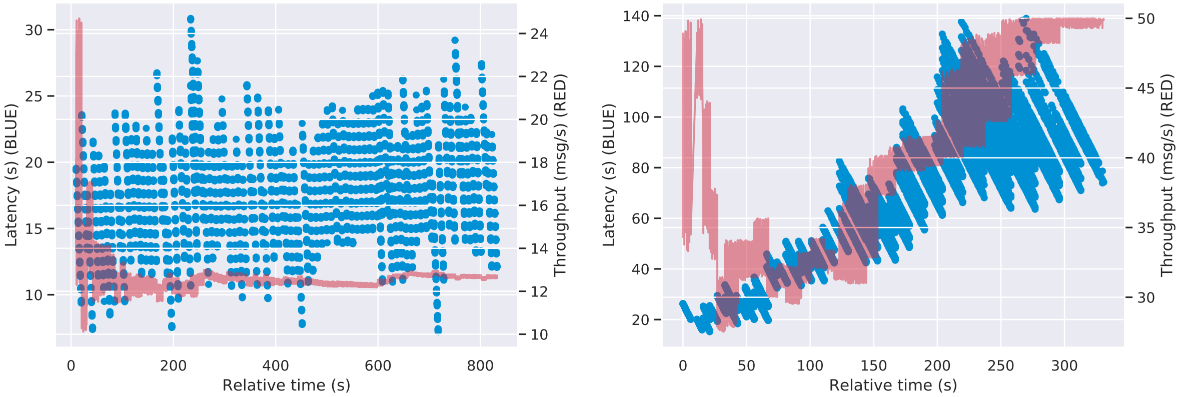


Figure 4.19: Throughput and latency of a 3-peer mix-net (preliminary results on developer laptop) with input rates of 12 (left) and 24 (right) msg/sec. Note that the mix-net on the right fails after around 300 seconds since it cannot process the incoming messages fast enough.

In summary, the preliminary tests show that we can continuously run the mix network with over 12 and up to 20 messages per second, which amounts to between just over 1M and 1.7M messages per day. As positive result, this already confirms that we can achieve the lower limit of the KPI of 1M messages per day on an ordinary developer laptop.

4.3.2 Full-scale Experiments

Our preliminary tests already confirmed that we can easily reach the minimal KPI of 1M messages per day on a developer laptop. However, we also wish to reach the upper KPI limit of 5M messages per day (≈ 60 messages per second), which seems out of reach with a normal laptop computer. For the full-scale experiments, we therefore migrate the mix-net to the cloud and run it on a dedicated server powered by two Intel Xeon E5-2670 CPUs with 8 cores (16 threads) each and a total of 256GB of RAM.

We use Panoramix to setup and run several mix-net deployments with configuration parameters as follows:

- We run mix-nets with **between 2 and 10** mix servers (*number of peers*).
- We configure the *batch size* to **50, 100, or 300** messages before mixing starts.
- We vary the *input rate* **from 12 to 144** messages per second.

After setting up and starting the mix network for each configuration, we run a simulation that injects messages from several clients into the network at the specified overall input rate. For each message, we keep track of the time when it was sent. On the receiving end, we can then compute the latency between sending and receiving each message, as well as the total throughput (output rate) of the system.

4.3.3 Results

Figure 4.20 illustrates the overall results of the experiments. We discuss each parameter in turn.

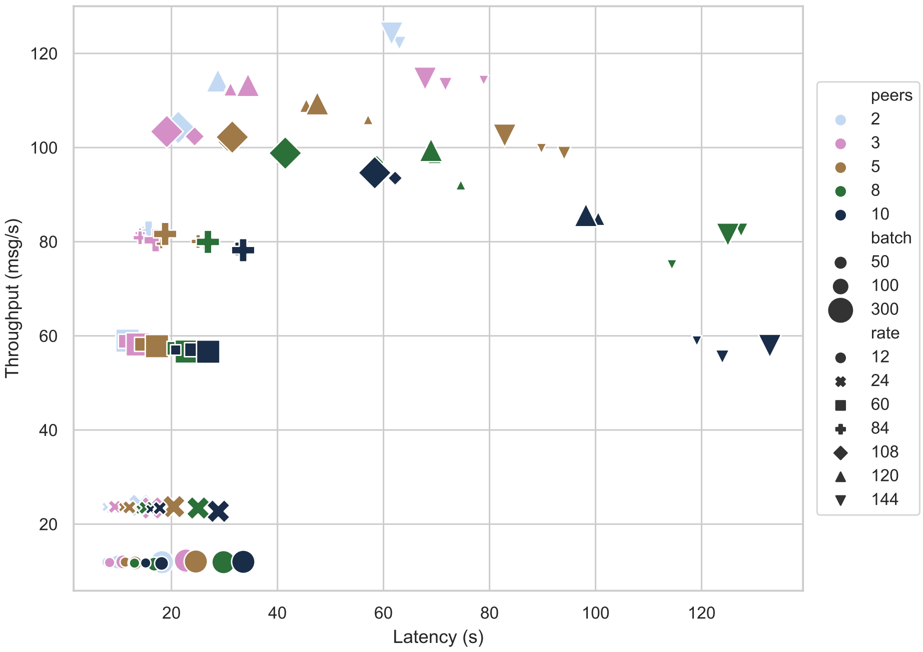


Figure 4.20: Effects of number of peers, batch size, and input rate on latency and throughput.

Number of peers Since each peer requires a certain amount of time to process an incoming batch of messages, each additional peer will add latency to the system. On the other hand, assuming that each peer is fast enough to handle a certain message throughput, adding another peer will only have little to no *direct* effect on the overall throughput. However, each peer also needs to communicate with the mix-net controller and hence will demand extra computing time from the server for each message that it needs to process. Therefore, we hypothize that if the server is already running near its maximum capacity, additional peers will affect the overall performance.

As we can see, this is confirmed in figure 4.20: For each given batch size and input rate, increasing the number of peers almost always increases the overall latency (from bright to dark plot markers). Furthermore, as can be seen in figure 4.21, the system easily can handle up to 100 messages per second even with 10 peers. For those input rates the overall throughout across different numbers of peers remains largely unchanged. But if we send more than 100 messages per second, additional peers will not only add latency, but also more and more decrease the overall throughput of the system.

Batch size The batch size is a configuration parameter of the mix network and specifies the minimal number of messages that a node has to receive before it starts mixing and passing on the messages. Intuitively, larger batch sizes induce more latency since it takes longer until enough messages have been received at each node. This is confirmed in figure 4.20, where bigger batch sizes (larger markers) typically show larger latencies (further to the right) than smaller batch sizes.

On the other hand, our results show only limited impact of the batch size on throughput; in particular for lower input rates there is barely any noticable effect. However, for very high input rates of 120 msg/s and above, and numbers of peers that the system can still handle (2, 3, 5), we see that larger batch sizes can provide slightly better performance; we assume this is due to more efficient handling since fewer batches need to be processed if they contain more messages.

Input rate The input rate controls the rate at which messages are fed into the network and theoretically directly relates to the overall throughput of the system. However, mixing and relaying the messages through the mix-net is computationally intensive and therefore puts a limit on the maximum rate at which messages can be transported. We can observe this in figure 4.20, where the throughput increases with the input rate, until around 108-120 messages per second. This is a very good result compared to our developer laptop, since this shows that we can easily reach the upper limit KPI of 5M or even 10M messages daily (amounting to 60 or 120 messages per second). Also note that the latency remains below 40s for all tested configurations with input rates up to 84 msg/s.

If we keep increasing the input rate after 108 or 120 msg/s, the throughput starts decreasing again, for large-peer networks first. The 10 and 8 peer networks (black and green markers) reach their peak throughput first at $\sim$ 108 msg/s, followed by the 5 peer network at $\sim$ 120 msg/s. Smaller networks with 2 and 3 peers *seemingly* continue increasing their throughput even when sending up to 144 msg/sec. However, similar to our the preliminary experiments on the developer laptop, even those smaller networks eventually collapse from congestion under such high loads as we will examine below.

Consider figure 4.21 (left), where we again plot latency and throughput for a 3-peer mix-net with batch size 100. Both throughput and latency generally grow with the input rate. After some time, the network throughput approaches the desired input rate as long as it is not too high (here: below 100). However, when we send 144 messages per second, although the input rate keeps increasing at first, also the latency builds up and eventually the network breaks down after around 300 seconds, even though we are only running 3 peers. For a larger networks, the collapse happens earlier and also even with lower input rates, cf. figure 4.21 with a 10-peer network shown on the right.

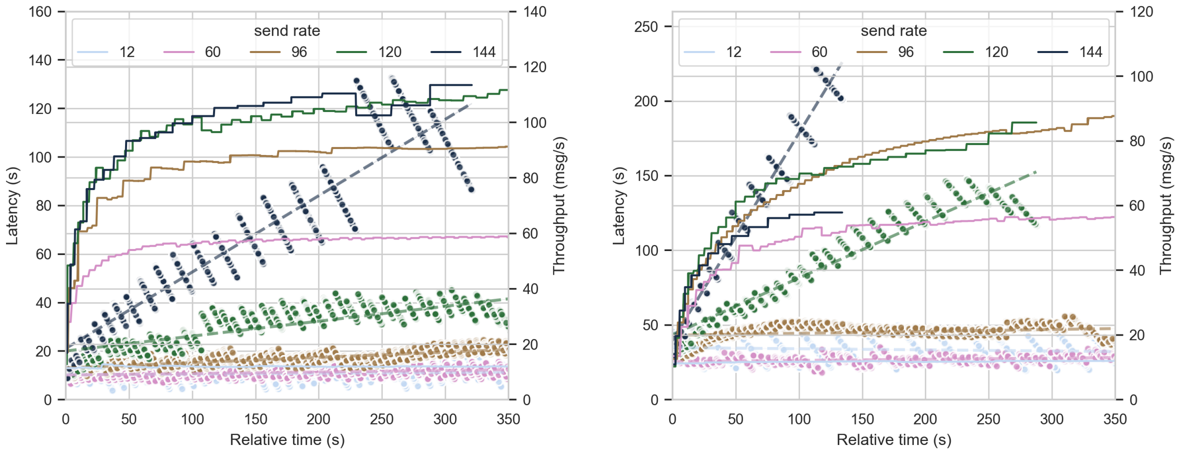


Figure 4.21: Evolution of latency (scatter plot) and throughput (solid lines) for different input rates on a 3-peer mix-net with batch size 100 (left) and a 10-peer mix-net with batch size 300 (right).

Processing more messages also takes more computation time at each node and at the controller, hence increasing the input rate also comes with an increase in latency. For smaller input rates that the system can process fast enough this increase in latency is negligible. However, if the input rate becomes too high, the mix-net can suffer from congestion and collapse, as discussed above.

Last (but not least), there is one interesting effect with large batch sizes that can be observed e.g. in figure 4.21: Contrary to the trend of larger input rates giving slightly larger latencies, a very low input rate of 12 msg/s produces *higher* latency than an input rate of 60 msg/s. The reason is that with low input rates and large batch sizes the batches fill less quickly, and hence messages have to wait longer until they can be processed.

5. Discussion

While our validation and testing section gives an impression of the current coverage of the qualitative requirements ANA-R1, ANA-R2 and CLI-R3, it also shows that it is depending on a variety of specific parameters. However, the evaluation metrics also provide a foundation for discussion and eventual consensus between data analysts, data owners and data curators. Furthermore, the validation confirms that a sweet spot between utility and privacy can be achieved, and that both anonymization strategies make a valuable contribution by complementing their strengths.

As a general remark our religion attack model underlines that meta-data attacks (i.e., network information) are mitigated by PANORAMIX mix networks. In addition, we considered a membership inference adversary with some prior knowledge by building a membership inference attack model. The PL mechanism provides strong protection against the MI attack when $\epsilon \leq 0.7$. In respect to utility, we conducted an analysis in form of the Jaccard distance and earth mover's distance allow us to infer that meaningful utility is achievable under the privacy level $\epsilon \geq 0.5$. Taking together the insights on utility and privacy for the Planar Laplace mechanism, we conclude that the trade-off between privacy and accuracy is possible for $0.5 \leq \epsilon \leq 0.7$.

5.1 Lessons Learnt and Guidance to Future Adopters

Selection of DP algorithm While rigorous anonymization approaches such as differential privacy allow quantification of the privacy loss (e.g., ϵ), the concrete choice of an anonymization mechanisms among multiple options, and the interpretation of the obtained (allowed) privacy loss can be hard.

Within this PANORAMIX use case we addressed this by analyzing the change in concrete utility and privacy in the form of business analytics and attacker simulations for several privacy losses. This dual approach allowed us in to search for a sweet spot where utility is still meaningful in the context of business applications and privacy is still strong enough against specific attackers. For the mobility datasets evaluated herein we found such a sweet spot. However, we recommend to repeat these analytics and check which analytics are most relevant for utility and which attackers are most threatening to the users.

Furthermore, the use case once more underlined that there are scenarios where cryptography and anonymization complement each other. While anonymization with differential privacy is allowing us to specify and enforce quantifiable privacy guarantees directly the data source, the mix network ensures secure & anonymous transfer between the data source and the data collection server. Both approaches individually would have fallen short to realize this use case.

Selection of mix-net parameters There are several parameters involved for setting up a mix network that influence its privacy protection characteristics as well as its performance, in particular latency and throughput.

A central idea of mix networks is to obscuring network traffic and thus protect privacy by having each message traverse through several *mix-net nodes*. To a small extend, the number of mix-net nodes also affects latency as each peer in the route from sender to receiver adds some delay to the communication. We found in our experiments that as long as the message rate is sustainable (i.e. each peer is fast enough to process the expected message throughput, and the mix-net controller has enough resources to handle the communication overhead for the additional peer) the increase in latency by additional peers is negligible. As guideline for a business deployment, the number of peers should at least match the number of business partners involved to enforce separation of power.

Another major parameter is the *batch size* (number of messages before mixing is started) which influences performance and privacy protection. Smaller batch sizes reduce latency since mixing operations can begin faster, which may be important if there are response time or (near) real-time requirements. However, this may come at the cost of throughput (messages per second) since more mixing operations cause more computational overhead in the mixers and the mix-net server. More importantly, larger batch sizes are also beneficial as they provide better privacy protection through longer delays and by chunking messages into larger batches. We therefore favour larger batch sizes for extra privacy as long as latency requirements permit.

Last but not least, the *input rate* directly relates to the maximum throughput of the system. However, it is capped by the available computational power from both mix nodes and the mix server (controller). By running the controller on a dedicated server with 16 worker processes, we could scale the mix-net to sustainable throughputs of up to 108-120 messages per second (depending on the number of peers), amounting to up to 10M messages per day and outperforming the desired KPI of 5M messages daily. We advise that the required throughput of a system must be determined in advance so the hardware for running the mix-net can be aligned with these performance requirements and prevent the network from congesting and collapsing.

6. Conclusion

This report summarized the validation and testing results of our demonstrator for privacy-preserving data collection and statistics. This includes the verification of requirements that were identified in the first two years of the project, the evaluation of privacy gains, effects on utility, as well as the impact on performance associated with the use of two complementing privacy-enhancing techniques: Mix networks help to protect communication metadata, such as IP address or message timestamps, and differential privacy helps to protect sensitive data values themselves.

Our experiments demonstrated that for our use case in WP6 of PANORAMIX, a sweet spot between privacy and utility exists. However, it is required to contextualize the differential privacy guarantee through concrete attacks and utility metrics to find sweet spots and communicate guarantees to end-users. Similarly, we recommend to also consider the evaluation of concrete attacks as well as performance requirements when selecting the mix network parameters.

Bibliography

[DR13] Cynthia Dwork and Aaron Roth, *The Algorithmic Foundations of Differential Privacy*, Foundations and Trends in Theoretical Computer Science **9** (2013), no. 3-4, 211–407.

[Eve02] Brian Everitt, *The cambridge dictionary of statistics*, Cambridge University Press, 2002.

[Hit41] Frank L Hitchcock, *The distribution of a product from several sources to numerous localities*, Journal of mathematics and physics **20** (1941), no. 1-4, 224–230.

[Jac01] Paul Jaccard, *Étude comparative de la distribution florale dans une portion des alpes et des jura*, Bull Soc Vaudoise Sci Nat **37** (1901), 547–579.

[Kos16] Sven Kosub, *A note on the triangle inequality for the jaccard distance*, arXiv preprint arXiv:1612.02696 (2016).

[Pra] *New york city prayer times*, http://www.andresmh.com/nyctaxitrips/.

[PTDC17] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro, *Knock knock, who's there? membership inference on aggregate location data*, arXiv preprint arXiv:1708.06145 (2017).

[PW08] Ofir Pele and Michael Werman, *A linear time histogram metric for improved sift matching*, Computer Vision–ECCV 2008, Springer, October 2008, pp. 495–508.

[RTG98] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas, *A metric for distributions with applications to image databases*, Computer Vision, 1998. Sixth International Conference on, IEEE, 1998, pp. 59–66.

[WB17] Benjamin Weggenmann and Daniel Bernau, *Interim report - Deliverable D6.1*, PANORAMIX Project, 653497, Horizon 2020 (2017).

[ZFX^{+}11] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li, *Geolife gps trajectory dataset - user guide*, July 2011.