DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Crowdsourcing mobility data with privacy preservation through decentralized collection and analysis

Simon van Endern

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Crowdsourcing mobility data with privacy preservation through decentralized collection and analysis

Crowdsourcing von Mobilitätsdaten ohne Einschränkung der Privatsphäre durch dezentrales Sammeln und Analysieren

Author:	Simon van Endern
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	Trinh Viet Doan
Submission Date:	30.06.2019

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 30.06.2019 Simon van Endern

Acknowledgments

Abstract

We propose a method to publish location ata without raising privacy concerns.

As still this data could be useful for many stakeholders, we will investigate how on the one hand aggregated data can be published without imposing any privacy risk to the owners of the data and on the other hand develop a prototype of a mobile application through which this location data is aggregated in a decentralized manner so that the raw user data never leaves the users' device.

Contents

Acknowledgments **iii**

Abstract **iv**

1 Introduction **1**
1.1 Motivation 1
1.1.1 General motivation: We want to open-source data 1
1.1.2 Main problem: Conflict between privacy and publishing user data 1
1.1.3 Existing privacy problem: The availability of central data sets . . 1
1.1.4 Relaxed privacy problems: The limited but not eliminated risk through a modified central data set 1
1.1.5 New problem in case of modified central data set: Trust to the server is needed 2
1.2 Research Question: No central raw data set but only aggregated data . 2
1.3 Contributions 2
1.4 Outine 3

2 Related Work **4**
2.0.1 Classification of location data and apps that use it 4
2.0.2 What has been achieved so far 4
2.1 Inferring data from already published datasets 4
2.2 Summaries of papers 8
2.2.1 Approaches to avoid central datasets 11
2.2.2 Infer activities from location data (and publicly available data) . 15
2.2.3 Crowdsourcing 15

3 Solution **17**

4 Design **19**
4.1 Overall Design 19
4.1.1 Android Application 19
4.1.2 Server application 20
4.1.3 Public database 20

4.2 Specific designs 21
4.2.1 Standard user story 21
4.2.2 Data aggregation schemes 22

5 Conclusion **24**

List of Figures **25**

List of Tables **26**

Bibliography **27**

1 Introduction

1.1 Motivation

1.1.1 General motivation: We want to open-source data

"Data is the new oil" is an often quoted stigma and means that more and more businesses are based not on specific production capacities but on data, the ability to process it and the exclusive owenership over it. The success and monopoly of companies like Google or Facebook can be attributed to this exclusive ownership to a significant extend.

According to commonly accepted economic theories [TODO: quote / reference exactly], monopolies hinder innovation and progress. This implies that the unavailability of huge amounts of data to the public is an impediment of innovation and increased growth. Nevertheless, the publication of raw data sets is impossible because it severly intrudes the privacy of the owners of the data.

1.1.2 Main problem: Conflict between privacy and publishing user data

Thus we see a conflict between preserving user privacy and publishing user data.

1.1.3 Existing privacy problem: The availability of central data sets

Nevertheless, user privacy is already compromised even if without publication of user data.

Already the mere existence of central data sets pose a privacy risk to users, because security issues might allow for theft and unwanted publication of these data. An example is the facebook data scandal representative for many data breaches over the last years. TODO: [Find and cite].

1.1.4 Relaxed privacy problems: The limited but not eliminated risk through a modified central data set

Some governments and other institutions already publish some of their data sets after anonymizing them e.g. through cloaking of data so that it achieves k-anonymity

and there are crowdsourcing and open source approaches to make data available to everybody. Nevertheless, the applied anonymization is often not sufficient or at least critical if the resulting data set should still be useful. Research shows that inferences can be drawn from the published data sets that violate the respecitve users' privacy. So, in addition to the main risk of a central data set, publishing anonymized data poses another risk to users privacy.

1.1.5 New problem in case of modified central data set: Trust to the server is needed

Furthermore, besides the remaining risk of inference attacks in published anonymized data sets, the anonymization through those algorithms always depend on a trusted server to collect the data from all users and then publish the results of any analysis applying privacy-preserving algorithms beforehand. So even if the data is only stored anonymized on the server, besides the remaining risk of inference attacks, this still imposes a high privacy risk to every user, as trust can be misued by the trusted server itself.

1.2 Research Question: No central raw data set but only aggregated data

Clearly, in order to overcome the conflicit between privacy intrusion and (public) data availability, a solution is needed that gets along without storing raw data in a central data set. This solution should 1. eliminate the risk of leaking raw user data through theft from a centralized database and 2. eliminate the remaining risk of inference attacks on published believed-to anonymized raw data. So far, we have not seen an approach to fully solve this problem.

1.3 Contributions

For our solution, we will focus on the sub-area of location data and location privacy. We investigate the possibility of storing raw location data only decentralized on the collecting devices. On a central server available to the public, only aggregated data is stored, thus the main problem of privacy risk by a central database containing the overall raw data set is solved. Furthermore, the issue of trust is removed, as the aggregation process happens decentrally, thus the central server will never hold any other data than aggregated data. It will never know about the individual raw data.

In summary, our approach takes the opposite direction as todays standard. We do not first collect the whole data set and then reduce it to a data set meeting privacy-constraints but we start from the bottom up - first by performing analysis in a decentralized manner so that there never is an overal data set imposing a security risk on all the entries' users, and second by proposing a framework that only releases aggregated data where no interference of any user information is possible. This data will then be available to the public. This gives us maximum possible feedback on eventual privacy problems, creates trust through transparency and fosters innovation through availability of data to everyone.

1.4 Outine

The structure of our research is organized as follows: First we re- view related work in the areas of location privacy and anonymisation techniques. In section 3 we describe our approach of decentralized data analysis to get along without a central database. Section XXX describes the setup in detail. Section XXX analysis the result from field-testing our application. Section XXX incorporates the results into our proposal of a possibility to achieve 100% privacy through all applications. Section XXX summarizes our work and points out further research possibilities.

2 Related Work

2.0.1 Classification of location data and apps that use it

In order to review existing approaches and research, classify location aware services by the acceptable delay of the location information being available:

- Almost no delation tolerance: e.g. an application showing a pop-up about a nearby venue e.g. a coffe shop when a pedestrian passes
- Some delay e.g. one minute is acceptable: An application e.g. google maps derives the information of congested traffic from devices reporting their GPS data which show lower than usual speed. As congestions worth reporting last longer than one minute, some delay in the device's information reaching the server is acceptable.
- Significant delay of hours, days or even weeks is acceptable for historical and statistical use of location data e.g. to find out about popular visiting times

2.0.2 What has been achieved so far

Most existing approaches focus on publishing location data where a huge delay is acceptable as can be seen in the following table: TODO [create table].

2.1 Inferring data from already published datasets

- privacy issues
  - Centralized databases also expose the users to a security risk (through theft) [29, 19].
  - Research has shown, that even from a location data set that is pseudonymous, i.e. the identifiers have been stripped or anonymized from the data, it is still possible to infer the home location of single users through inference attacks [23, 12, 15, 19, 36]. The same problem arises when using data collected through crowdsourcing [22].

  - Furthermore, this location coordinates can then be combined with publicly available information e.g. reverse map coding of coordinates to addresses and then searching for entries in telephone books to infer the users identity from it's home location [23, 15, 19]. This identity can then be linked to other sensitive data. This problem also arises in the area of IoT [29, 19].
  - Often (though with usually lower probability) also the work address in addition to the home address can be inferred and makes linking the data to identities even easier [12, 15].

- solutions
  - Spatial cloaking: Achieving k-anonymity by dropping data points or perturbing them or dropping all data points around a random point around the home location [23].

- Problems when applying those solutions.
  - Insufficient accuracy / the data set becomes useless [23, 12, 27, 32, 31].
  - Extending the time period over which data is collected generally increases the risk.
  - Anonymization techniques might score well in densly populated areas or areas with high traffic but poorly in sparsely populated areas especially where a single address can be mapped to a single person or family [20, 1, 19] [location-privacy correct paper or cited wrong paper???] or might not work for individuals whos work and home location are further away than average [15].
  - Still all approaches depend on first centrally collecting the original raw data and then before querying [32] applying anonymization techniques.
  - Data suppression algorithms have only limited success and can only reduce, but not eliminate the risk [19].

- More soffisticated approaches
  - [20]

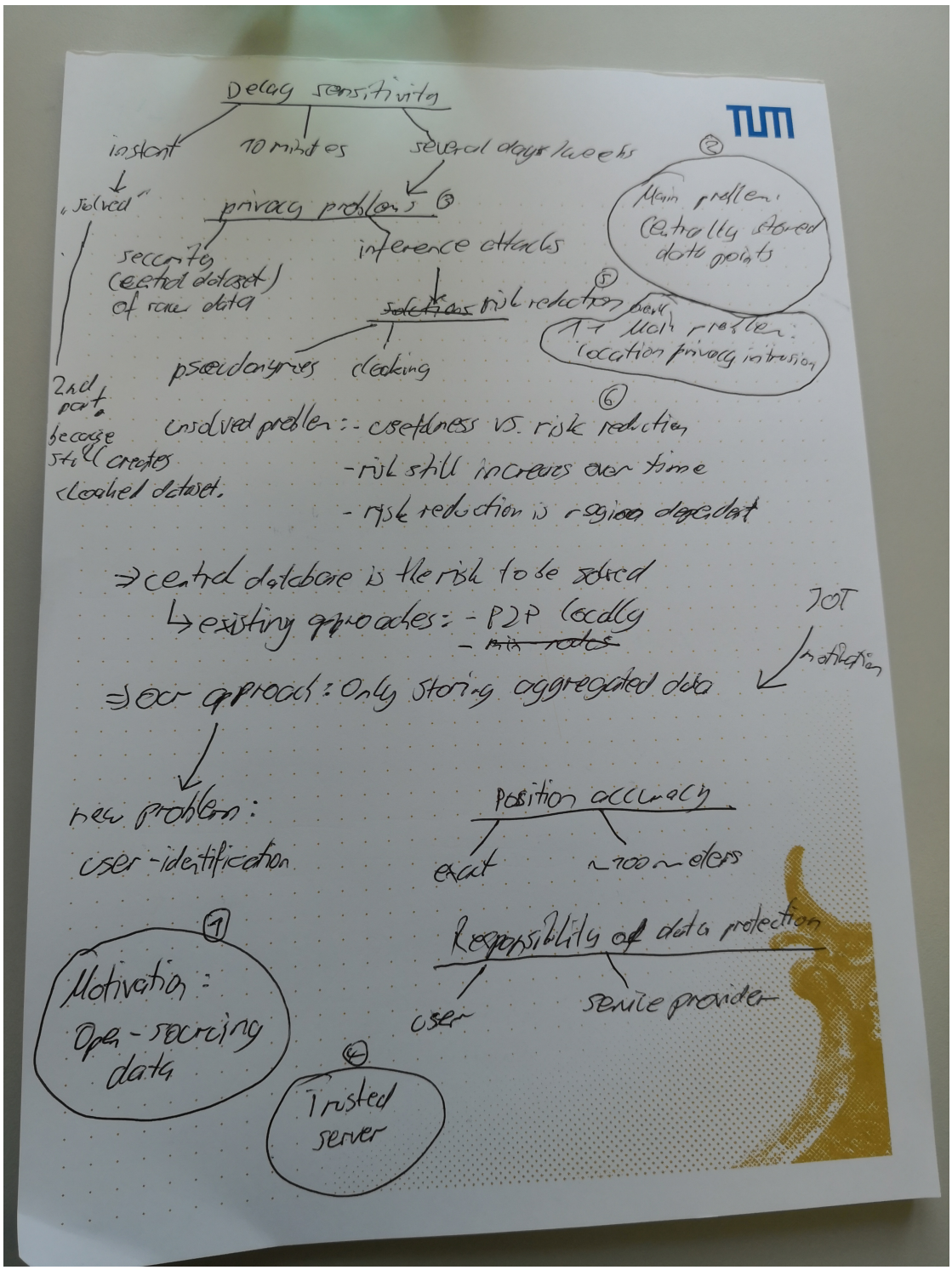
- Approaches tackling instant data use
  - [1, 2] introduces mix-nodes, that can nevertheless not guarantee privacy and also depnds on a trusted third party. Also [26] proposes a solution (close to our summary) how to enable privacy for instant use of location data.

- Solutions to overcome a central database.

  - [21] proposes the use of P2P over WIFI and Bluetooth to decrease the need of central instances.
  - [22] proposes a secure approach where the raw data is hidden from the central instance but still the aggregated data can be obtained by using encryption methods. This approach is very close to our work. Also [19] is close to our work and uses encryption.
  - [19] proposes an approach to handle user authentication.

- Decentralized methods for data analysis are also motivated from the area of IoT [29].

TODO: Relate to [32]


Delay sensitivity
instant
10 minutes
several days/weeks
„solved"
privacy problems ③
security (central dataset) of raw data
inference attacks
~~solutions~~ risk reduction ⑤
pseudonymes
cloaking
② Main problem: Centrally stored data points
⑤ 1-1 Main problem: location privacy intrusion
2nd part because still creates cloaked dataset.
⑥ unsolved problem: - usefulness vs. risk reduction
- risk still increases over time
- risk reduction is region dependent
⇒ central database is the risk to be solved
↳ existing approaches: - P2P locally
- ~~min nodes~~
⇒ our approach: Only storing aggregated data
IOT motivation
new problem: user-identification
Position accuracy
exact
~100 meters
Responsibility of data protection
user
service provider
① Motivation: Open-sourcing data
④ Trusted server

2.2 Summaries of papers

[23] is one of the first investigating privacy issue in location data. For inferring the home location of a set of car travelling traces of their research subjects, they identify taking the last destination of the car before 3 pm as the most successful among 4 algorithms / heuristics to determine a persons home location. They where able to identify 12.8% of the users home coordinates. Furthermore by looking up those home coordinates on a free online tool, they are able to retrieve the correct name for about 5% of the subjects. Nevertheless, those results could be improved by far, as they show that the used data source / white pages are outdated. In order to protect anonymity, they mainly identify the following different countermeasures:

- Pseudonymity: Stripping original IDs from the dataset (by many shown not to be sufficient)
- Spatial Cloaking: Application of k-anonymity by hiding all data points in a circle with the center placed randomly around the actual home address
- Noise: Adding Gaussian noise to each data point
- Rounding: Placing a grid on the location data and mapping each data point to the closest intersection
- Dropped Samples: Completely dropping samples in order to reduce the frequency of the data points.

Of the application of these countermeasures, only spatial cloaking can preserve data quality, while a Noise with a standard deviation of 5km and a grid for rounding with 5km distances is needed, which render the data useless for many applications. On the contrary, they showed how easy it is, to make the final step from home location to actual identity. Furthermore, there analyzis is only based on data covering two weeks.

[12] finds that even when personal data is anonymized thus that names and addresses, etc. are removed, sensitive information can be inferred from the data. In this study it was shown that from call-records in the US the home address and also often the work address of a person could be inferred. They highlight that while adhering to the k-anomymity model proposed by [32] it is practically not possible to publish datasets that are still of any significant use.

Also [15] highlights the thread that home and work locations can be inferred from anonymized datasets and can in combination with other sources yield even more information about a user. To reduce this risk, they propose "to collect the minimum

amount of information needed". In contrary, we want to investigate another approach, so that rich data can still be used and be published in an aggregated manner to let people profit from the data but still preserve privacy.

Another problem that arises is that anonymization algorithms applied to datasets prior to publishing them might yield good results if the location data is in a densily populated area but might perform poorly if the population is only sparse [20].

[20] identify that while privacy algorithms might successfully provide privacy for location data samples in highly frequented areas, but perform poorly and disclose sensitive information for samples in areas with lower traffic frequency. They discuss the problem commonly accepted in research that either the quality of the data becomes poor or useless when applying techniques like k-anonymity [27, 32, 31] or that privacy cannot be guaranteed. They propose a novel algorithm based on time-to-confusion. Thus basically whenever it is possible to attribute two different samples of a dataset with a high probability to the same user, the corresponding sample gets removed from the data-set to be published. This is necessary, as "the degree of privacy risk strongly depends on how long an adversary can follow a vehicle" [20]. In more detail, time-to-confusion also takes into account the entropy information provided by the whole dataset, thus that even when two samples cannot be connected with high probability due to to many possible consecutive samples, analyzing the whole dataset can provide information that actually the possible consecutive samples have different probabilities due to common route choices. E.g. a vehicle on a highway is much more likely to follow on the highway for some more time than leaving the highway. While this information is taken into account, they point out the limitations of their work that when the dataset is matched with street maps, even more samples would have to be remoed to ensure privacy because it will render some former possible consecutive samples impossible due to missing streets connecting them.

[1] introduces the concept of mix-nodes already known from privacy research on a network level (TODO: "copy" related work part of paper "time-to-confusion"). They propose a framework in which privacy is protected through frequently changing pseudonyms. Furthermore they find that similarly to the problem of identifying consecutive samples in [20], the change of pseudonyms has also to be obfuscated in order to provide complete privacy. In contrast, this paper focuses mostly on solving the problem that location aware services that e.g. notify you when you are close to a venue of interest, do not need to have access to your location data at anytime but can register to events with a mix-node. Thus they register for the venues of interested and only get notified when the mix-node, which is trusted and has complete access to location data, detects a match. One sees straight away, that this again depends on trust of the users on the mix-node. Nevertheless, the proposed solution of mix-nodes and mix-zones analyzed on a sample shows that even using this framework, privacy cannot

be provided, especially as here again the entropy provided by the history of the released or somehow collected data-set makes it too hard to obfuscate the consecutiveness of different pseudonyms.

[32] is the current state of the art of minimum data protection. They define a dataset as the commonly understood tables in SQL. Besides the unique identifier used in the table, a quasi-identifier is the combination of several attributes with which a set of entries can be identified. a dataset adheres to the rules of k-anonymity, if querying every possible such identifier returns at lest a set of k different entries. Thus 1-anonymity identifies an entry exactly and provides no anonymity at all. The anonymity problem arises not from the dataset itself, but from a combination of datasets, that have the attributes of the quasi-identifier in common. This way anonymous knowledge from both datasets can be linked in order to infer information not intended to be made public. They also highlight, that also publishing the same dataset with different privacy-rules, i.e. different anonymization techniques applied, can result in inferences that reveal the original dataset.

[32] clearly highlights that there are two approaches to hiding sensitive information. One is to restrict queries to a database that might reveal sensitive information. In contrast to this approach, they focus on anonymizing the data already before any access to it. Nevertheless, this is based on the assummption that the data owner knows about any possible quasi-identifier in order to obfuscate the dataset sufficiently to provide k-anonymity for all quasi-identifiers. If one quasi-identifier is not thought of, the dataset might expose 1-anonymity for this identifier and result in possible exposures of data not intended to be public.

[32] also discusses further problems that are easy to tackle but nevertheless necessary to protect users' privacy. The order of the published table must be random. Otherwise there is more information (hidden) available that can be used to break k-anonymity. Another problem is when the same table is released and obfuscated differently for the same quasi-identifier, other attributes in the releases can be used to link entries and thus de-anonymize the data.

[15] further investigates the fact that from a dataset containing GPS data of trajectories or e.g. twitter-posts as in [36] the home location can be inferred with high probability. They show that also the work location can be identified with pretty high accuracy and probability. Furthermore they find that people who live and work in different regions or more generally, the further work and home diverge, the smaller the anonymity set of the specific user in the dataset and thus the lower also the anonymity. This is similar to the findings of [1] that users in less populated areas are exposed to more privacy risk than in denser areas.

[2] extends the analyzis of [1].

TODO: Cite approach of disclosure algorithms by [17] TODO: Cite confusion ap-

proach similar to [20] by [18] TODO: Read [33]

2.2.1 Approaches to avoid central datasets

[21] addresses the possible solution of p2p communication instead of using a central instance. They state, that mobile p2p communication is mainly based on WIFI and bluetooth. They propose a middleware embedded on top of the android operating system to facilitate widespread use of p2p. However, those p2p networks are so far limited to devices close to each other locally, as it works over WIFI or Bluetooth and so far there is no established approach to connect smaller local p2p networks over the internet to completely stop relying on central server instances.

[22] investigates the problem that in crowdsourcing (with real humans) the reported results, thus the work of the workers allows for inferences about personal information of the (anonymous) worker. To achieve this, they use decentralized computation while "guarantee[ing] security of the proposed protocol". They use as an example the case where a map of publicly available automated external defibrillators is generated through crowdsourcing. Through the reported location data, the privacy of the "worker can be invaded". The state, that so far, only a few have investigated the privacy problems in crowdsourcing (like [3]). The use a "sum protocol" where sums can be added in an enccrypted way and then be decrypted finally. This is based on the work of [10]. [25] also find, that this sum protocol is the only way (in crowdsourcing) to guarantee privacy, as otherwise messages between workers would have to be exchanged (our telegram, etc. approach.) In this paper, two other papers are mentioned, that deal with aggregating data while preserving privacy: [6], [28].

[29] motivates decentralized data storage and anyalysis as well, though stemming from another motivation. In IoT, bandwith limits require devices to perform decentral analysis and only forward aggregated data to a central database or to aggregate data in a completely decentralized manner. Others, that deal with decentralized analysis are [4] and [35]. Furthermore, similar to our motivation, they find that "Especially sectors that deal with highly personalized information, such as healthcare, require according means for the secure and privacy-preserving processing of data". Apparently also [34] and [30] find, that inferring information from data (not location data in this case) is possible. (TODO! read the papers if its correct). Another related source for distributed analysis is [11]. They define the term "fully decentralized" as where information is only exchanged with local peer nodes. No coordinator is needed thus. Also this paper highlights, that the centralization of data at one single point poses a high risk in case of data theft and furthermore is a single point of failure. They quote from [7] that the most power intensive thing in smartphones is receiving and sending data. For vertically distributed data, as it is also the case for our mobility data, they find five different modi

of data analysis:

- Central analysis
- Local preprocessing, central analysis
- Model consensus
- Fusion of local models: Each local node builds its own model. Those models are transmitted to a coordinator and fused to a global model
- Fusion of local predictions: When a prediction is made, predictions of the single nodes are collected and combined into a consensus

[19] also tackle the problem that through to "inference attacks" users can be reidentified even if the location data are anonymized. Especially home locations are easy to infer in suburban scenarios where one person / family belongs to one address. Such sensitive information as home, medical visits, etc. might be inferred. The are also aware that using geo-coded address databases it is easy to infer a person's identity from it's home location. The further quote [9] (TODO!!) who state that for obtaining "traffic flow information" it is sufficient, if 5% of all traffic participants send data. To ensure anonymity of data owners, they propose a system in which a trusted third party interacts as mediator between the data owners and the traffic server. The location data is encrypted with the traffic servers public key (private, public key pair) and sent to the third party, using a symmetric encryption algorithm. This way the third party server can confirm the identity of the sender and thus prevent fraud, while it has no access to the data itself, which it forwards to the traffic server. This is a huge advance, but still it is based on a trusted third party server. Also it depends on savely storing (and putting) the symmetric keys into the car / mobile device. They further find that it is necessary to sanitize the incoming data on the traffic server because it might be simply wrong or malicious. This is also necessary in our approach, but in general only, if data is sent as a push variant by devices. In a pull scenario, the data could only be compromised, if somebody manages to attack the application itself. And this can still be handled by sending the aggregation request to different groups in parallel and see whether one request delivers outliers. In addition, they are aware, that the data stored at the traffic server still bears a privacy risk if access to this data is obtained. Thus our approach to totally decentralize the collection and analysis is a huge and necessary step forward. This is especially even more compromising, if one takes road maps into account in order to identify the home locations they find as well. Furthermore, the mapping of a location to an identity is facilitated by white pages like telephone books or real estate records they state. They also report manual inspection in order to map locations to

homes which is easy as for most areas there are precise satellite images which help to outrule possibilities. They manage to identify about 27% of plausible home suggestions from a 239 record data set. (Not validated as exact home location is not revealed in the data set used). Even when data samples are dropped so that there is one GPS point every 10 minutes instead of every minute, still more than 16% of plausible home locations can be detected. So data suppresion algorithms do reduce the risk of home identification but only to limited extend.

[26] tackles the common problem through introduction of a "location anonymiszer" and a "privacy-aware query processor" which enables a user to set its privacy to match k-anonymity (k is variable) and also to choose the degree of spatial-cloaking applied to its data. This approach also requires a trusted third-party (the location anonymizer). Mainly the concept works as following: The third party server receives the exact location data of every user and for each data point creates a new data point through spatial-cloaking that satisfies its privacy setting which then is stored on the server. Whenever a location service (note this approach is mostly for the location-services requiring instant or medium delay data) needs a users location e.g. to notify him about near gas stations (private query), the server receives only the obfuscated location data (a region) and answers with a list of possible matches within this region which is than on the end users device matched with the exact location to retrieve e.g. the nearest gas station. The second benefit is that applications can still reach out to users in a specific area (e.g. to send a promotion to all users in a certain area) through sending this request to the third party which then also replies with a list / area of end users that respects all privacy settings. this is called public query. (On page 764 there might be additional sources - TODO!!). They say that closest to their work is [14] (TODO!!!) and [16] (already read and summarized). Though this still poses the problem of a centralized data set at the query processor. Even the data is pseudonymous and locations are only blurred, other papers like [TODO: cite the others] revealed that even when blurring location data, inference of home locations is still possible. Especially as XXX states, when the data considers rather sparsely trafficed areas. Furthermore identification attacks become more and more precise, the more data points are available (bigger time-span). They further prove, that the overhead of sending a list of possible matches as response to a private query is acceptable (if the data privacy setting is not to strict i.e. $k < 50$ and cloaked region size < 64 in one region).

[16] investigates privacy preservation through spatial and temporal cloaking also using a (third party) middleware and a generated location data set. They furthermore talk about the topic that the participant itself should be anonymous (network identifier). This can be achieved in our setting through forwarding a request instead of adding data or sending it to the database in a percentage of cases. They state that this has been investigated by [8] and also many papers refer to onion routing regarding this

topic. Also the message size is always padded in order to hinder inferences. They differentiate location services according to three measurements:

- Frequency of Access
- Time-accuracy / Delay sensitivity
- Position accuracy

There motivation stems from using sensors already installed in cars instead of separately equipping roads with sensors. This highlights also that our approach is not only limited to location data itself but can be generalized (though usually all information is related to location and time). They state that that for traffic / road information usually high accuracy of the information is not necessary and also a few minutes of delay are tolerable. This highlights, that our approach is also feasible for traffic data and not only pure historical data. They make the example that e.g. harsh breaking data might be used to infer bad road conditions and warn other traffic participants. Regarding this data, huge delays are acceptable as usually a dangerous crossing stays dangerous for a long time (if not forever). They also state that for retrieving nearby points of interest, the time accuracy is high, thus results are needed instantly, but location accuracy does not need to be high. This is in line with the findings of [26]. Additionally, they classify threats to privacy into two categories:

- Sender anonymity (which can be tackled by us by forwarding with a certain probability)
- Inferences of repeated samples (which we tackle by not publishing raw data)

The solve this problem by using a (third-party) anonymity server which on the one hand acts as a mix-node and thus solves problem one (also by pseudonyming the data) and also applies cloaking in order to solve the second problem (partially as shown in other papers). They use the definition of anonymity according to k-anonymity (page 35) [Good definition]. They follow an approach that takes time and space into account. If not enough users for k-anonymity where in a certain area, they either apply spatial cloaking (making the data less accurate) or temporal cloaking (gathering in a certain area for a longer time). Also a mixture is possible we think, though then one has to take into account that the same data is not included in different publications. This is also thematized by them using an example and stated that tuples must not overlap in time and space. They find that in their experminental setting, a temporal cloaking of 70 seconds or a spatial one of 250 meters is sufficient (in their setting!) to provide some feeling about the extends. Definition of security as the successfull attempt of an adversary to retrieve data not intended to be public / "violate anonymity constraints".

They also talk about the possibility that a user spoofs the service by fake users. We also will address this problem to limit fake users.

2.2.2 Infer activities from location data (and publicly available data)

[24] investigates the possibility to label certain times as specific activities (at home, at work, shopping, dining out, visiting, all other) through the use of location data (reduced to the binary variables near restaurant and near store), supervised machine learning and publicly available information of places, etc. They use machine learning and relational markov networkds for it. Also they take transitions into account, e.g. that one does not go from dining out to dining out or also one does usually only dine out a maximum number of times a day. They find that when using data from different subjects, the algorithms can be trained to have an error rate below 20% when labelling activities, in one experimental setting they even achieve an error rate of 7%. This shows, that using open source software to infer activities from location data should be pretty accurate right now (the paper is 14 years old right now) so when we infer activities like walking, driving, on a bus, etc. We can assume a pretty low error rate which is in favor of the correctness of our analysis's.

2.2.3 Crowdsourcing

[5] clarifies the definition of crowdsourcing, especially in contrast to open source (which makes all the components of a product available and free to use - which is especially easy with software and trickier with hardware), presents some of the first big crowdsourcing approaches and draws conclusions. Crowdsourcing is defined as the distributed outsourcing of work to single individuals. Nevertheless, in contrast to open source, the collected and aggregated result will be property of the company, the individuals only get a compensation. The state that following another definition, crowdsourcing applies only, if the result is later on used in mass production. They also talk about the downside - namely that the companies make huge profit through crowdsourced ideas and the bounties distributed to the workers are a very low compensation regarding the success of the company and far less than full-time employees would usually be paid. [This would be different in our approach as it combines crowdsourcing and open sourcing]. Crowdsourcing is indeed not only used for simple but also for complex problem-finding tasks. They highlight, that it is hard to make a living from open source, as it does not pay a share to the contributers. They further highlight the limits of crowdsourcing (especially through the internet) as the typical user is around 30 years old, English speaking, western world. This highlights that crowdsourcing should be considered very carefully when information other than pure facts should be generated.

Also for example when considering a road-map of where people walk, it might show some paths as not used at all, while many older people prefer those paths - but do not carry a reporting device / smartphone.

As a final learning one can take away that similar as in Linux, a small trustworthy core like our proposed application would solve many problems due to its verifiability.

3 Solution

We will use the definition of location privacy as defined by [1]: " the ability to prevent other parties from learning one's current or past location". They further propose a different approach to preserve privacy. TODO!!!

We develop a framework ...

Beyond the scope of this research is ...

Possibilities for the decentralized analysis are:

- Send data via an application e.g. whatsapp or telegram, that takes care of message encryption (trusted third party)
- use encryption methods that allow for aggregation as in [22]

A possiblility to infer information about the whole crowd from just a small subset is investigated by [13]

Use data formats according to the platform, that shows the standard for each type of data. (Naming conventions, find this website open data, ... something like that)

Blockchain might be useful in this area as well.

What furthermore all papers do not take into account is that when the overall time of sampling increases, the precision of inference attacks will automatically increase as well.

TODO: Search for papers regarding inference attacks on pure aggregated data

What kind of spatial cloaking to apply? City, region, country? Or purely geometric?

Important: If spatial cloaking request for aggregating data is not successful, it has to be discarded and then retried with a coarser area, but no two overlapping results must be published.

Address the problem of fake users (submitting fake data).

It also seems possible to do the following:

- Store the whole set of venues, etc. of the users home area on the device (in a coarse level). This way the users local data does not have to be revealed at all (and storage space on modern (!!!) phones is sufficient.) It can also be limited / rule based e.g. max. 1 GB data.

- install software that directly cloaks the GPS location of the smartphone on the device (and labels it as cloaked). This way services accepting cloaking can respond to this and set a list instead of a single result, services not adhering to this policy will cease to work properly.
- Only allow our application to collect (centralized location data).
- Limit every applications sending capabilities to a standardized format for authentication (and also for max x times per hour in order to prevent misuse of this template for still exporting data). Most applications do not even require an account to work properly and those who do, can use this template. (e.g. to send emails. For registration purposes an exception has to be implemented). If the application depends on location data, it has to use (and also improve, thus the motivation is there) the publicly available data set.

This way we can ensure that either any service only gets a data set of cloaked regions or no data set at all and provide 100% data privacy. Especially this approach empowers the user, thus the decision to stay private is not anymore based on the goodwill of the service provider but put into the hands of the user itself.

Say how many of the google play store's top 20 applications for location could work without sending any information to the service itself or where inaccurate location data would be sufficient.

For aggregating data as necessary for google road maps, the hugher the passed on data set becomes, the more dangerous? Because it temporarily is the same as a central server, so malicious users could extract the information from the application.

TODO: cite openStreetMap project as open source, open sourcing data.

Create label for privacy / internationally recognized brand so that applications having this sign are trusted.

4 Design

4.1 Overall Design

Our architecture comprises the following:

1. An Android Application
2. A server application
3. A public database (with a visualizing website)

4.1.1 Android Application

The Android Application needs the following:

1. Create a public/private key pair
2. Automatically collect Location Raw Data for the scheme Timestamp, Location –> TODO: Implement database in Android Application
3. Create activities from the location raw data for the scheme From, To, Duration, type, ...
4. For each day? week?, ... compute the areas where the user was active.
5. Stage 2: (or even stage 1?): After each day, send anonymously, thus over several nodes, in which area the device has been active! (that is a breach of our privacy!!! find solution) in order for the server to know to whom send requests regarding locations.
6. Send registration request to server application
7. receive Aggregation request from server application
8. calculate response to aggregation request
9. Apply logic when to abort aggregation request e.g when incoming n is 0 and next device ID is empty.

10. send aggregation response
11. At least one screen displaying some text / information about the application
12. Automatic hard-coded push notification to check for an update at date xxx
13. Automatic hard-coded deletion of all data after test-phase including a push notification to ask for the deinstallation of the app.

In stage two, the App will also be able to send e.g. traffic alerts to the server.

4.1.2 Server application

The server application needs the following:

1. A database listing all registered devices following the scheme public device key, Google Cloud messaging key (for reaching the device).
2. A database containing all possible aggregation requests
3. A scheduler to start / send out aggregation requests
4. A handler for an ongoing aggregation request (forward it to the next one, until done or aborted).
5. Store a requests result in the public database

In stage 2, the server also processes data send by the client on its own behalf. The server aggregation task does the following

- ~~Randomly select a fake-start-n (out of the range of lets say 1-5) in order~~ With encryption not necessary
- ~~Select fake-start-value~~ Not necessary with encryption
- Devices list for the aggregation request (Will be made dynamic in stage 2).

4.1.3 Public database

The public database comprises the following schemes for aggregation requests. Furthermore, it handles incoming data e.g. traffic alerts. The aggregation schemes will all contain at least the following fields:

- Current n

- Current mean / value **Use json (or xml) for value passing**
- ID ?? (necessary?)
- Next device's public key for encryption of the data (and n?)

4.2 Specific designs

4.2.1 Standard user story

Our user is called Hans

1. Hans somehow gets motivated to go to the playstore and install our appliation
2. In the playstore, Hans sees some photos and information about the application
3. Hans clicks on the install button in the playstore to install our application. The installation process starts.
4. Hans is curious about the application and opens the application.
5. Hans sees the first and only screen of the application that tells hime what the application does. It also contains a link to view the results stored in the public database.
6. (In stage 2, maybe Hans can even see his data and what has been send, ...)
7. Hans leaves the application (the application must still go on in the background).
8. Hans uses his task manager to quit the application (the application must still go on collecting data in the background).
9. After some days, without Hans being involved, an aggregation request is started and send to the appliation running in the background.
10. The application receives and processes the request and sends the results to the server without Hans noticing anything.
11. One week after the installation, tha application creates a push notification and asks hans to update the application. It also says that the update is less than 1MB and asks him to install it right now, as it is so few data and in order not to forget to do it later
12. Hans updates the application and thus installs all the fixes we have done in the meantime.

13. After the end of the testing period, the application automatically deletes all data. Hans gets shown a push notification informing about this and is asked to deinstall the application. A thank you is displayed as well.

4.2.2 Data aggregation schemes

We will in stage one ask all devices and only in stage 2 allow for limiting to specific areas. The minimum n is always 5 (choosen out of "Bauchgefühl"). TODO: evaluate which n values are necessary. When computing average and median, n is at least 10 (Bauchgefühl). When reporting a full box-plot, n is at least 20 (Bauchgefühl). For reporting skewness and standard deviation as well, 50 is the minimum for n (Bauchgefühl).

We definitely need a strict approach for overlapping, etc. to also take care of cases we have not thought about.

TODO: From overlapping, ... it will be possible, to compute "There is somebody, generylly not walking, but biking, in this area, ... ". Check, whether part of this can somehow be used as a quasi identifier.

Average walking, driving, ... time

For each of the activities [walking, biking, driving a car, driving public transport] a request is emitted to compute average data. The request can further for each activity exclude or include 0-value-computation. Other solution: the 0-values are counted and the average / median of the other data (only in case).

How many people go to work by car, ...

Need sophisticated methods to determine work time, ...

How many people combine bike with public transport

select everybody where activity biking and activity public transport are close to each other.

Create a road map

Aggregate all trajectories of the users longer than xx meter, cap the ends and the start for 50 meter, put similar trajectory coordinates together and publish the whole map. A users data is only included, if in a radius of xx there are xx more –> k-anonymity. Create a car / bike / walking map or color-code the map.

Compute the average speed (talking daytime into account) for roads

This can e.g. be used to identify roads where zone 30 will reduce noise and co2 exhaustion a lot, if one can compute, that cars accelerate and then typically rest again, so that an average of 30 would have the same timing result. [TODO: Cite cities, ... where 30 is the limit in citycenter, ...]

Time to work

Compute, how much time on average a user needs to go to work. This requires implementing locally the calculation of a users work and home place from the data (or we could also ask for the data and state that it will be locally only) or combine the two approaches.

Average time at work

How much time to people usually spend searching for a parking spot

Identify roads, where (maybe even including the current traffic light sequence) one would be faster with a bike (thus average speed of 20, 22, ...)

Stage 2: When moving on a specific road, report bad traffic

We will fake this in a way that when somebody is driving (recognized activity), we will randomly trigger the application to send an alert. This way we do not have to implement downloading the standard speeds for a road map, recognizing a user being on this road and then checking for less than usual speed.

5 Conclusion

TODO: Hint, that our approach can be easily integrated in other applicaitons like open maps projects.

Relate to open street maps project introduced in [23]

List of Figures

List of Tables

Bibliography

[1] A. R. Beresford and F. Stajano. "Location privacy in pervasive computing." In: *IEEE Pervasive computing* 1 (2003), pp. 46–55.

[2] A. R. Beresford and F. Stajano. "Mix zones: User privacy in location-aware services." In: *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*. IEEE. 2004, pp. 127–131.

[3] M. Bernstein, E. H. Chi, L. Chilton, B. Hartmann, A. Kittur, and R. C. Miller. "Crowdsourcing and human computation: systems, studies and platforms." In: *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2011, pp. 53–56.

[4] S. Bin, L. Yuan, and W. Xiaoyi. "Research on data mining models for the internet of things." In: *2010 International Conference on Image Analysis and Signal Processing*. IEEE. 2010, pp. 127–132.

[5] D. C. Brabham. "Crowdsourcing as a model for problem solving: An introduction and cases." In: *Convergence* 14.1 (2008), pp. 75–90.

[6] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. "SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics." In: *Network* 1.101101 (2010).

[7] A. Carroll, G. Heiser, et al. "An Analysis of Power Consumption in a Smartphone." In: *USENIX annual technical conference*. Vol. 14. Boston, MA. 2010, pp. 21–21.

[8] D. L. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms." In: *Communications of the ACM* 24.2 (1981), pp. 84–90.

[9] X. Dai, M. A. Ferman, and R. P. Roesser. "A simulation evaluation of a real-time traffic information system using probe vehicles." In: *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*. Vol. 1. IEEE. 2003, pp. 475–480.

[10] I. Damgård and M. Jurik. "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System." In: *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*. PKC '01. London, UK, UK: Springer-Verlag, 2001, pp. 119–136. ISBN: 3-540-41658-7.

[11] K. Das, K. Bhaduri, and H. Kargupta. "A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks." In: *Knowledge and information systems* 24.3 (2010), pp. 341–367.

[12] K. Drakonakis, P. Ilia, S. Ioannidis, and J. Polakis. "Please Forget Where I Was Last Summer: The Privacy Risks of Public Location (Meta)Data." In: *CoRR* abs/1901.00897 (2019). arXiv: 1901.00897.

[13] S. Ertekin, H. Hirsh, and C. Rudin. "Learning to predict the wisdom of crowds." In: *arXiv preprint arXiv:1204.3611* (2012).

[14] B. Gedik and L. Liu. *A customizable k-anonymity model for protecting location privacy*. Tech. rep. Georgia Institute of Technology, 2004.

[15] P. Golle and K. Partridge. "On the Anonymity of Home/Work Location Pairs." In: *Proceedings of the 7th International Conference on Pervasive Computing*. Pervasive '09. Nara, Japan: Springer-Verlag, 2009, pp. 390–397. ISBN: 978-3-642-01515-1. DOI: 10.1007/978-3-642-01516-8_26.

[16] M. Gruteser and D. Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking." In: *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM. 2003, pp. 31–42.

[17] M. Gruteser and B. Hoh. "On the anonymity of periodic location samples." In: *International Conference on Security in Pervasive Computing*. Springer. 2005, pp. 179–192.

[18] B. Hoh and M. Gruteser. "Protecting location privacy through path confusion." In: *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. IEEE. 2005, pp. 194–205.

[19] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. "Enhancing security and privacy in traffic-monitoring systems." In: *IEEE Pervasive Computing* 5.4 (2006), pp. 38–46.

[20] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. "Preserving Privacy in Gps Traces via Uncertainty-aware Path Cloaking." In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*. CCS '07. Alexandria, Virginia, USA: ACM, 2007, pp. 161–171. ISBN: 978-1-59593-703-2. DOI: 10.1145/1315245.1315266.

[21] W. A. Jabbar, M. Ismail, and R. Nordin. "Peer-to-peer communication on android-based mobile devices: Middleware and protocols." In: *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*. IEEE. 2013, pp. 1–6.

[22] H. Kajino, H. Arai, and H. Kashima. "Preserving worker privacy in crowdsourcing." In: *Data Mining and Knowledge Discovery* 28.5-6 (2014), pp. 1314–1335.

[23] J. Krumm. "Inference Attacks on Location Tracks." In: *Pervasive Computing*. Ed. by A. LaMarca, M. Langheinrich, and K. N. Truong. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 127–143. ISBN: 978-3-540-72037-9.

[24] L. Liao, D. Fox, and H. A. Kautz. "Location-Based Activity Recognition using Relational Markov Networks." In: *IJCAI*. Vol. 5. 2005, pp. 773–778.

[25] X. Lin, C. Clifton, and M. Zhu. "Privacy-preserving clustering with distributed EM mixture modeling." In: *Knowledge and information systems* 8.1 (2005), pp. 68–81.

[26] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. "The new casper: Query processing for location services without compromising privacy." In: *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment. 2006, pp. 763–774.

[27] P. Samarati and L. Sweeney. "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and specialization." In: *Proceedings of the IEEE Symposium on*. 1998.

[28] A. Shamir. "How to share a secret." In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[29] M. Stolpe. "The internet of things: Opportunities and challenges for distributed data analysis." In: *ACM SIGKDD Explorations Newsletter* 18.1 (2016), pp. 15–34.

[30] M. Stolpe and K. Morik. "Learning from label proportions by optimizing cluster model selection." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2011, pp. 349–364.

[31] L. Sweeney. "Achieving k-anonymity privacy protection using generalization and suppression." In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 571–588.

[32] L. Sweeney. "k-anonymity: A model for protecting privacy." In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[33] K. P. Tang, P. Keyani, J. Fogarty, and J. I. Hong. "Putting people in their place: an anonymous and privacy-sensitive approach to collecting sensed data in location-based applications." In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2006, pp. 93–102.

[34] S. Thrun, L. K. Saul, and B. Schölkopf. *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. Vol. 16. MIT press, 2004.

[35] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang. "Data mining for internet of things: A survey." In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pp. 77–97.

[36] H. Zang and J. Bolot. "Anonymization of location data does not work: A large-scale measurement study." In: *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM. 2011, pp. 145–156.