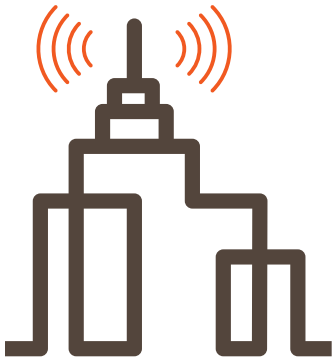
VaVeL
H2020 - 688380



VaVeL

D4.1 - Report on Data Privacy

TU Dortmund University

May 30, 2017

Status: Final

Scheduled Delivery Date: May 31, 2017

Document History

- (December 15, 2015) First Version, Structure of Deliverable
- (March 25, 2017) Major Content Update
- (May 19, 2017) Finalization of V1
- (May 24, 2017) Revision According to Internal Review Comments

Executive summary

This document will be a report on Task 4.2 "Privacy Issues of Citizen generated data" under the description of WP4. In this document we discuss the privacy threats in spatio-temporal citizen data processing and present methods that enable privacy preserving data analysis. Thus, we present the contributions of the consortium on privacy preserving analysis (including amongst others also traffic prediction, crowdsourcing, data gathering). In addition, this deliverable provides a state-of-the-art survey on privacy preserving analyses to assist other researchers with privacy preserving data analysis for citizens.

Please check the website of the project (`www.vavel-project.eu`) under the deliverables section for additional deliverables and updates.

Document Information

Contract Number	H2020-688380	**Acronym**	VaVeL
Name	Variety, Veracity, VaLue: Handling the Multiplicity of Urban Sensors		
Project URL	http://www.vavel-project.eu/		
EU Project Officer	Mr. Francesco Barbato		

Deliverable	**D4.1**	Report on Data Privacy		
Work Package	**Number**	WP4		
Date of Delivery	31/05/2017	**Actual**	31/05/2017	
Status	Final			
Nature	Report			
Distribution Type	Public			
Authoring Partner	TU Dortmund University			
QA Partner	Fraunhofer IAIS			
Contact Person	Thomas Liebig	thomas.liebig@tu-dortmund.de		
	Phone		**Fax**	

List of Contributors: Thomas Liebig (TUD), Katharina Morik (TUD), Marco Stolpe (TUD), Gennady Andrienko (Fraunhofer IAIS), Ioannis Katakis (UoA), Dimitrios Gunopulos (UoA), Vana Kalogeraki (AUEB).

Project Information

This document is part of a research project funded by Horizon H2020 programme of the Commission of the European Communities as project number 688380. The Beneficiaries in this project are:

No.	Name	Short Name	Country
1	National and Kapodistrian University of Athens	UoA	Greece
2	Technische Universität Dortmund	TUD	Germany
3	Technion - Israel Institute of Technology	Technion	Israel
4	Fraunhofer-Gesellschaft Zur Förderung Der Angewandten Forschung E.V.	Fraunhofer	Germany
5	IBM Ireland Limited	IBM	Ireland
6	AGT International	AGT GROUP (R&D) GMBH	Germany
7	Orange Polska S.A.	OPL	Poland
8	Dublin City Council	DCC	Ireland
9	City of Warsaw	CoW	Poland
10	Warsaw University of Technology	WUT	Poland

Table of Contents

Index of Figures

1 Introduction

This deliverable serves as a report on the task 4.2 in Work Package 4, "Data Quality, Privacy and Crowdsourcing". This document is focusing on the single task of 'privacy preserving data analysis', in contrast to the other deliverables that comprise multiple tasks. Thus, we organized this deliverable by structuring the existing literature on privacy-preserving data analysis methods and highlighting our contributions in this field. With this structure, the deliverable not only reflects the achievements of the VaVeL consortium driven by the requirements, but the literature survey itself is a contribution to research and application of privacy preserving citizen data analysis in other projects.

Whenever citizen data is *stored*, *transmitted* or *analysed*, the privacy of the citizens may be affected. In this deliverable we contribute with privacy preserving methods for the following use cases:

- **Crowdsourcing** using entropy based filtering,
- **Distributed traffic prediction** with learning from label proportions,
- **Trajectory aggregation** with homeomorphic encryption,
- **Data recording** with episodic movement data.

In general, there are two classes of privacy-preserving data-mining methods. We distinguish between "random perturbation-based" and "secure multiparty computation".

The random perturbation based methods modify the data, i.e., additive or multiplicative noise is added to the data. The original, vulnerable, properties of the data are protected, but data mining may still operate on this data if it does not differ too much from the original data set. A sample for a random perturbation based method is [LKR06]. In their work, the authors show that random projection based multiplicative data perturbation provides an efficient method to perform privacy-preserving data mining. In general, random perturbation based methods exchange quality of the analysis results with the gained data privacy. Secure multiparty computation, on the other hand, involves multiple parties, and the operations are performed without revealing vulnerable data to other parties. This does not generate any trade-off among quality of the analysis and privacy, as no data is modified.

In this deliverable, we do not just highlight the contributions of the VaVeL project to both class of methods but also provide a brief overview on the field of privacy-preserving data analysis and integrate our methods into the existing state-of-the-art literature.

1.1 D4.1 and its connection with other deliverables

Figure 1 illustrates the connections of this deliverable with the rest of the VaVeL deliverables. These relationships are briefly described bellow:

- The techniques described within this document have to match the requirements set from the two use cases. The use cases are introduced in **D7.1** (Dublin use case) and **D8.1** (Warsaw use case).

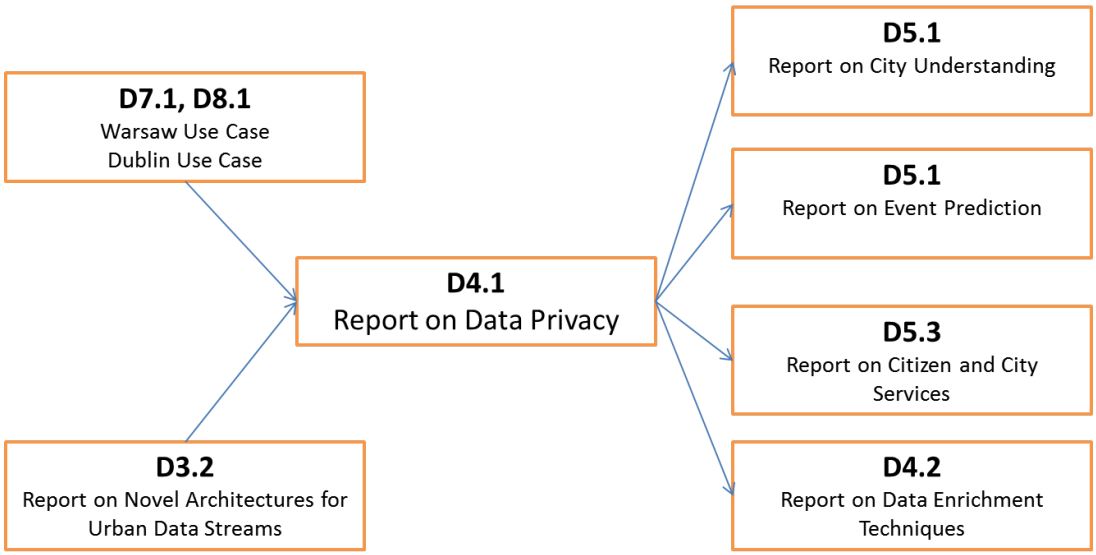


Figure 1: Relationship of D4.1 with other deliverables.

- Since the data enrichment techniques **D4.2** require the processing of sensitive data, they should take advantage of the methods described here.
- The methods described in D4.1 address storage, transmission and analysis of heterogeneous spatial data. Thus, it influences **D3.2** "Report on Elastic and Resilient Infrastructures", in order to effectively use the available infrastructure.
- All methods that are (or will be) described in **D5.1**, **D5.2** and **D5.3** can benefit by the privacy-preserving analysis techniques described in this deliverable.

1.2 Relevant VaVeL Publications

Publication	Section
[Lie17] T. Liebig. Smart navigation - chances, risk and challenges. In M. Jankowska, M. Pawelczyk, S. Augustyn, and M. Kulawiak, editors, *Navigation and Earth Observation - Law & Technology*, page (in press). IUS PUBLICUM, Warsaw, 2017	Sec 2.3 Sec 3.1
[BK16] I. Boutsis and V. Kalogeraki. Location privacy for crowdsourcing applications. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 694–705, New York, NY, USA, 2016. ACM	Sec 4.4
[Lie16] T. Liebig. Ai-based analysis methods in spatio-temporal data mining. In M. Jankowska, M. Pawelczyk, S. Allouche, and M. Kulawiak, editors, *AI: Philosophy, Geoinformatics & Law*, pages 133–150. IUS PUBLICUM, Warsaw, 2016	Sec 2.3
Publications of the VaVeL consortium in preparation for the VaVeL project:	
[LSM15] T. Liebig, M. Stolpe, and K. Morik. Distributed traffic flow prediction with label proportions: From in-network towards high performance computation with mpi. In G. Andrienko, D. Gunopulos, I. Katakis, T. Liebig, S. Mannor, K. Morik, and F. Schnitzler, editors, *Proceedings of the 2nd International Workshop on Mining Urban Data (MUD2)*, volume 1392 of *CEUR Workshop Proceedings*, pages 36–43. CEUR-WS, 2015	Sec 3.2
[SLM15] M. Stolpe, T. Liebig, and K. Morik. Communication-efficient learning of traffic flow in a network of wireless presence sensors. In *Proceedings of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases (PDCKDD 2015)*, CEUR Workshop Proceedings. CEUR-WS, 2015	Sec 3.2
[Lie15b] T. Liebig. Privacy preserving centralized counting of moving objects. In F. Bacao, M. Y. Santos, and M. Painho, editors, *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, pages 91–103. Springer International Publishing, 2015	Sec 3.4 Sec 6.1 Sec 6.2
[Lie14] T. Liebig. Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services*, pages 86–99, 2014	Sec 3.4 Sec 6.1 Sec 6.2
[BK13] I. Boutsis and V. Kalogeraki. Privacy preservation for participatory sensing data. *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 0:103–113, 2013	-
[AGDG+13] G. Andrienko, A. Gkoulalas-Divanis, M. Gruteser, C. Kopp, T. Liebig, and K. Rechert. Report from dagstuhl: the liberation of mobile location data and its implications for privacy research. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(2):7–18, 2013	Sec 2.3
[AAS+12] N. Andrienko, G. Andrienko, H. Stange, T. Liebig, and D. Hecker. Visual analytics for understanding spatial situations from episodic movement data. *KI - Künstliche Intelligenz*, pages 241–251, 2012	Sec 3.1

Table 1: Relevant VaVeL publications to this deliverable.

1.3 Research Questions Answered

The research questions are derived from the requirements of the use cases. However, we reflect state-of-the-art methods on privacy preserving analyses and provide in Sections 5 and 7 also methods which are not necessarily suitable for our use-cases but may be relevant to the interested reader.

ID	Research Question	Conclusions	Sec
Q1	Which privacy threats are in spatio-temporal data?	Location data reveals lots of information on personal habits and preferences. Even data without personal identifier and aggregated data bear a risk of re-identification.	Sec 2
Q2	Can we effectively protect privacy in a centralized data set using aggregation methods?	Using episodic movement data individual movement is hidden in the aggregated values.	Sec 3.1
Q3	Can we use aggregated data in a distributed setting to predict traffic values at the sensor motes?	Communicating data or labels with the neighbors is sufficient for in-situ predictions. However, sending the proportions of labels the communicated data is reduced and individual privacy is protected.	Sec 3.2
Q4	Can we aggregate trajectories in a privacy preserving way at a centralized server without perturbing the aggregates?	Utilizing additive homeomorphic encryption scheme, data can be encrypted such that just the aggregation of k trajectories is available for analysis and no individual trajectories are revealed.	Sec 3.4, Sec 6.2
Q5	Can we protect individual privacy in crowdsourcing app?	Using filtering techniques, we can identify and omit linkable points in individual trajectories.	Sec 4.4
Q6	Is there a review of privacy preserving analysis techniques for spatio temporal data?	The deliverable at-hand provides a literature review of state-of-the-art methods for privacy preserving data storing, transmission and analysis techniques.	D4.1

Table 2: Research questions answered and main conclusions regarding this deliverable.

ID	Method Short Description	Approach	Sec
M1	Privacy Preserving Distributed Communication Efficient Traffic Flow Prediction	Learning from Label Proportions	3.2
M2	Privacy preserving centralizing trajectory data at centralized server	Paillier's Cryptosystem in combination with Shamir's secret sharing	6.2
M3	Private Sharing of Trajectories in Crowdsourcing Application	Entropy based Filtering of Data Items at Coresets	4.4

Table 3: Methods developed in deliverable D4.1.

2 Privacy threats in Spatio-Temporal Data

From a business perspective, mobility data with sufficiently precise location estimation are often valuable for enabling various location-based services; from the perspective of privacy advocates, such insights are often deemed a privacy threat or a privacy risk. Location privacy risks can arise if a third-party acquires a data tuple (user ID, location), which proves that an identifiable person has visited a certain location. In most cases, the datum will be a triple that also includes a time field describing when the person was present at this location. Although, in theory, there are no location privacy risks if the user cannot be identified or if the location cannot be inferred from the data, in practice it is difficult to determine when identification and such inferences are possible.

In the following we reflect an overview on privacy-aware learning by [WJD12]. In their survey they describe a long history of research at the intersection of privacy and statistics, going back at least to the 1960s, when Warner [War65] suggested privacy-preserving methods for survey sampling, and to later work related to census-taking and presentation of tabular data [GKS08]. The authors also highlight that there has been a large amount of computationally-oriented work on privacy [DMNS06a, Dwo08, ZWL08, WZ10, HRW11, DN03, BLR13, CMS11, RBHT09] which we are going to present and extend with our contributions in the following sections.

In our survey of possible privacy preserving techniques we highlight challenges for privacy-preserving analysis in real-time, and distributed applications and provide an overview over various privacy-by-design approaches:

- **Section 3:** Privacy via aggregation (with a contribution of the VaVeL consortium)
- **Section 4:** Privacy via data perturbation (with a contribution of the VaVeL consortium)
- **Section 5:** Privacy via sketches
- **Section 6:** Privacy via homeomorphic encryption (with a contribution of the VaVeL consortium)
- **Section 7:** Privacy via secret sharing

2.1 Terminology

In literature, the terminology for privacy is not used consistently, thus [PK01] proposed a common terminology of the privacy goals. In their work they distinguish amongst *Anonymity*, *Unlinkability*, *Unobservability* and *Pseudonymity* - terms that we also use within this document.

Definition 2.1 Anonymity *of a subject is the state of being not identifiable within a set of subjects (called the anonymity set) considering the information available to the observer.*

Definition 2.2 *The* Anonymity Set *is the set of all possible subjects. In case of spatio-temporal events, the anonymity set consists of all subjects that may have caused an event. In case of data recipients the anonymity set consists of all subjects who might be addressed.*

Thus, a person might be anonymous just within a set of potential data producers, his/her sender anonymity set, which may be a subset of all world-wide moving persons.

Definition 2.3 k-Anonymity *is guaranteed if a subject is not identifiable within any anonymity set of size k. [Swe02]*

Definition 2.4 *A data item is* indistinguishable *if it can not be distinguished amongst any other object of the anonymity set.*

Definition 2.5 Unlinkability *of two or more items (e.g. subjects, messages, events) means that it is impossible for the attacker, considering his knowledge of the whole system, to tell whether the two items are related or not. Put another way, the probability that two items are related stays the same before network setup (with the apriori knowledge) and after the run of the system (with the aposteriori knowledge).*

If we consider sending and receiving of messages from subjects (events, trajectories or spatio-temporal time series), anonymity may be defined as the unlinkability of a message to a subject identifier.

Definition 2.6 Unobservability *is the state of some spatio-temporal datum from a subject to be indistinguishable from any other datum at all.*

With respect to the same attacker, unobservability directly leads to anonymity.

Definition 2.7 Pseudonymity *is the use of pseudonyms as identifiers.*

By controlling collisions of the pseudonyms (e.g. using the same pseudonym more than once) and allowing partial reversal of the pseudonyms, pseudonymity comprises all degrees of linkability to a subject.

2.2 Threads from Moving Spatial Sensors

Recently, several location privacy incidents were reported in the media. A famous incident regards the case of Apple [Bil11a], where 3G Apple iOS devices were reported to store the location of their mobile users in unencrypted form for a period of over one year. This precise location information was stored without the knowledge of the users and was transmitted to the iTunes application during the synchronization of the device. According to Apple, the stored location information was not used to track the users but was attributed to a programming error which was later fixed with a software update. Google was also reported to be using precise location data, collected from users mobile devices, to improve the accuracy of its navigation services [Hel11], while Microsoft [McC11] recently admitted that their camera application in Windows Phone 7 ignored the users privacy settings to disable transmitting their location information to Microsoft. In response to this incident, the company issued a software update. Although the above-mentioned privacy incidents did not lead to actual harm caused to the individuals due to the lack of location privacy, the continual flurry of such breaches is worrying as it becomes evident that sensitive location information may easily fall into the wrong hands [Bil11b]. In the following subsections, we elaborate on different types of privacy risks that can lead to user identification or disclose sensitive location.

2.2.1 Collection of location information with assigned user ID

This is the most trivial case, as long as the location of the user is estimated with sufficient accuracy for providing the intended location based service (LBS). In cases where the location is not yet precise enough, various techniques (e.g. fusion of several raw location data from various sensors) allow for improved accuracy.

Example 2.1 *A cellular mobile network operator (MNO) generally stores tuples of the form (cell ID and sector ID, user ID), e.g. within the call detail records data (CDR) for billing purposes.*

Example 2.2 *A smartphone app gets the GPS-location for a user who has already been identified, e.g. by his/her login to the application or by a payment transaction.*

Example 2.3 *From a smartphone, a smartphone application provider receives the IDs and signal strengths of several nearby transmitters (base stations, WiFi devices, etc.). Based on previously established maps of these transmitters, the application provider is able to estimate a more precise location.*

Additionally, application providers may have direct access to a variety of publicly available spatial and temporal data such as geographical space and inherent properties of different locations and parts of the space (e.g. street vs. park) various objects existing or occurring in space and time: static spatial objects (having particular constant positions in space), events (having particular positions in time), and moving objects (changing their spatial positions over time). Such information either exists in explicit form in public databases like OpenStreetMap, WikiMapia or in smartphone application providers' data centers, or can be extracted from publicly available data by means of event detection or situation similarity assessment. Combining such information with positions and identities of users allows deep semantic understanding of their habits, contacts, and lifestyle.

2.2.2 Collection of anonymous location information

When location data is collected without any obvious user identifiers, privacy risks are reduced and such seemingly anonymous data is usually exempted from privacy regulations. It is, however, often possible to re-identify users based on quasi-identifying data that have been collected. Therefore, the aforementioned risks can apply even to such anonymous data. The degree of difficulty in re-identifying anonymized data depends on the exact details of the data collection and anonymization scheme as well as on the adversaries access to background information. Consider the following examples: Re-identifying individual samples. Individual location records can be re-identified through observation attacks [MYYR10, HQTK16]. The adversary knows that user Alice was the only user in location (area) l at time t, perhaps because the adversary has seen the person at this location or because records from another source prove it. If the adversary now finds an anonymous datum (l, t) in the collected mobility data, the adversary can infer that this datum could only have been collected from Alice and has therefore re-identified the individual. In this trivial example, there is actually no privacy risk from this re-identification because the adversary knew a priori that Alice was at location l at time t, so the adversary has not learned anything new. There are, however, three important variants of this trivial case that

can pose privacy risks. First, the anonymous datum may contain a more precise location l' or a more precise time t' than the adversary knew about a priori. In this case, the adversary learns this more precise information. Second, the adversary may not know that Alice was at l but simply know that Alice is the only user who has access to location l. In this latter case, also referred to as restricted space identification, the adversary would learn when Alice was actually present at this location. Third, the anonymous datum may contain additional fields with potentially sensitive information that the adversary did not know before. Note, however, that such additional information can also make the re-identification task easier.

Re-identification can also become substantially easier when location data is repeatedly collected and time series location traces are available. We refer to time series location traces, rather than individual location samples, when it is clear which set of location samples was collected from the same user (even though the identity of the user is not known). For example, the location data may be stored in separate files for each user or a pseudonym may be used to link multiple records to the same user.

Example 2.4 *A partner of a mobile network operator (MNO) has obtained anonymized traces of a user, e.g. as a sequence of call detail records (CDR) where all user IDs have been removed. While this looks like anonymous location data, various approaches exist to re-identify the user associated with these mobility traces. One approach is to identify the top 2 locations where the user spent most time. This corresponds in many cases to the user's home and work locations. Empirical research has further observed that the pair (home location, work location) is often already sufficient to identify a unique user [GP09].*

A recent empirical study [ZB11] explains various approaches for re-identification of a user. Another paper has analyzed the consequences for privacy law and its interpretation of increasingly strong re-identification methods [Ohm09]. Further re-identification methods for location data rely on various inference and data mining techniques. A recent study on cabs in the city of New York [DDFS16], analyses the potential risk of processing de-anonymised taxi trajectories.

2.2.3 Collection of data without location

Even in the absence of actual location readings provided by positioning devices, location disclosures may occur by means of other modern technologies. Recent work by Jun et al. demonstrated that the complete trajectory of a user can be revealed with 200m accuracy by using accelerometer readings, even when no initial location information is known [HON$^+$12]. What is even more alarming is that accelerometers, typically installed in modern smartphones, are usually not secured against third-party applications, which can easily obtain such readings without requiring any special privileges. Acceleration information can thus be transmitted to external servers and be used to disclose user location even if all localization mechanisms of the mobile device are disabled.

Another example of privacy disclosures in mobile devices regards the monitoring of user screen taps through the use of accelerometer and gyroscope readings. Recent work by [MVBC12] demonstrated that user inputs across the display, including the on-screen keyboard, of a mobile device can be silently identified with high precision through the use of motion sensors and machine learning analysis. Their prototype implementation achieved tap location identification

rates of as high as 90% in accuracy, practically demonstrating that malevolent applications installed in mobile devices may severely compromise the privacy of the users.

Last but not least, several privacy vulnerabilities may arise through the various resource types that are typically supported and communicated by modern mobile phone applications. [HHJ+11] examined several popular Android applications which require both internet access and access to sensitive data, such as location, contacts, camera, microphone, etc. for their operation. Their examination showed that almost 34% of the top 1100 popular Android applications required access to location data, while almost 10% of the applications required access to the user contacts. As can be anticipated, access of third-party applications to such sensitive data sources may lead both to user re-identification and to sensitive information disclosure attacks, unless privacy enabling technology is in place.

Example 2.5 *During a vacation, a user has taken many photographs, which are all tagged with a time-stamp but not geo-coded. There are, however, techniques to assign a geo-location to most images, as long as these they contain some unique features. Similarly, there are techniques to assign real names to most persons in the photographs, e.g. by using tools or crowdsourcing as provided e.g. by a social network or other platforms to store photos. Having times and places for a photo stream one might reconstruct precise trajectories.*

Example 2.6 *An app is able to continuously read the accelerometer of a handset. This enables it to reconstruct a 3D trace of the user's movements.*

2.3 Threats from stationary spatial sensors

Smartphones became a convenient way to communicate and access information. With the integration of GPS sensors, mobility mining was pushed forward [GP08]. The mobility information of multiple devices is usually stored on a server which performs analysis in order to extract knowledge on movement behaviour. In the easiest case this is the number of visitors to specific places. The processing of the data streams became unfeasible for large use cases, where millions of people are monitored and massive data streams have to be processed. In such Big Data scenarios, the expensive computation (matching and counting in individual, continuous GPS streams) is split among the parties and only the aggregation step remains on the server (In contrast [BK13] presents a method that distributes the query). Thus, continuous movement records (GPS) are reduced to episodic movement data [AAS+12] consisting of geo-referenced events and their aggregates: number of people visiting a certain location, number of people moving from one location to another one, and so on. The preprocessing of the GPS data streams is then performed locally on the location based devices and the aggregation is subject to crowd sourcing. Recent work focuses on in-situ analysis to monitor location based events (visits [KMM12], moves [HIJ+12]) or even more complex movement patterns [FMKM12] in GPS streams. In all cases a database with the locations or patterns of interest is provided in advance, and the mobile device computes event-histograms for succeeding time-slices. These histograms are much smaller and may be aggregated by the server in order to achieve knowledge on current movement behaviour. However, the transmission of such individual movement behaviour still poses privacy risks [AGDG+13, Lie16, Lie17]. Even access by third parties compromises individual privacy as disclosures on the NSA PRISM program reveal. The devices

monitor daily behaviour and thus reveal workplace and working hours, the place where users spent the night and other locations indicating information on sensitive subjects as health, religion, political opinions, sexual orientation, etc. Thus, the transferred episodic movement data may even lead to re-identification. The protection of the individual histogram in such a data stream of locally aggregated mobility events is therefore an important task. The adversary model is a compromised server that utilizes the received individual histogram for inference of identities and other sensitive data.

Example 2.7 *An intelligent traffic system that consists of stationary traffic loops records the number of cars per interval. At times with low traffic frequencies it might be possible to reconstruct individual trajectories.*

Example 2.8 *In case of additional data (e.g. CCTV) individual persons could even be identified.*

2.4 Strategies for Privacy Preserving Data Processing

Multiple methods to protect privacy are described in literature. They either operate at the network layer [KMM12] or, inspired by the differential privacy paradigm, they add random noise [MWP+13]. The work in [CKD+04] denotes a protocol for secure aggregation among multiple parties, but their algorithm requires extensive communication between the parties and is unfeasible in a single server scenario; also their encryption can be broken after several computation cycles. Alternatives are using sketches or aggregates of the data. Recently, [Lie15c] proposed usage of homeomorphic encryption for secure aggregation of distributed mobility histograms.

In [VS16] the strategies to ensure privacy in Big Data are systematically listed. The authors distinguish amongst eight strategies:

- **Minimize** individual data should be restricted to least possible quantity.
- **Hide** Private data must be concealed from unauthorized view.
- **Separate** Private data must be interpreted in separate partitions.
- **Aggregate** Private data should be treated with a better level of aggregation.
- **Inform** Data provider should be notified when their data is taken up.
- **Control** Data providers should have control over their data.
- **Enforce** a privacy strategy conforming to legal requirements should be used.
- **Demonstrate** Data managers must be able to validate privacy policy and any authorized actions.

In next sections, we will present several methods to ensure data privacy. In the summary (Section 8) we will provide an overview of the presented methods in this scheme.

More Information

T. Liebig. Smart navigation - chances, risk and challenges. In M. Jankowska, M. Pawelczyk, S. Augustyn, and M. Kulawiak, editors, *Navigation and Earth Observation - Law & Technology*, page (in press). IUS PUBLICUM, Warsaw, 2017

T. Liebig. Ai-based analysis methods in spatio-temporal data mining. In M. Jankowska, M. Pawelczyk, S. Allouche, and M. Kulawiak, editors, *AI: Philosophy, Geoinformatics & Law*, pages 133–150. IUS PUBLICUM, Warsaw, 2016

G. Andrienko, A. Gkoulalas-Divanis, M. Gruteser, C. Kopp, T. Liebig, and K. Rechert. Report from dagstuhl: the liberation of mobile location data and its implications for privacy research. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(2):7–18, 2013

This publications was published in preparation for the VaVeL project and has no VaVeL acknowledgement.

3 Privacy via Aggregation

At the first glance aggregation of data items seems to be a suitable way to ensure data privacy. Thus, in this section we address three aspects of aggregated data processing:

- data gathering,
- data analysis in distributed scenario and
- data analysis in a centralized scenario.

All sections reflect contributions by the VaVeL consortium.

3.1 Recording of Aggregated Movement Data

Most of the data collected by wireless sensor networks are referred to as "Episodic Movement Data": data about spatial positions of moving objects where the time intervals between the measurements may be quite large and therefore the intermediate positions cannot be reliably reconstructed by means of interpolation, map matching, or other methods. Three main types of uncertainty distinguish episodic from continuous movement data and these were identified in [AAS+12]. First, the most common type of uncertainty is the lack of information about the spatial positions of the objects between the recorded positions (*continuity*), which is caused by large time intervals between the recordings and by missed recordings. Second, a frequently occurring type of uncertainty is low granularity of the recorded positions (*accuracy*). Due to these two types of uncertainty, episodic movement data cannot be treated as continuous trajectories, i.e., unbroken lines in the spatio-temporal continuum such that some point on the line exists for each time moment. Third, the number of recorded objects (*coverage*) may also be uncertain due to the usage of a service or due to the utilized sensor technology. For

example, one individual may carry two or more devices, which will be registered as independent objects. Some recording techniques only capture devices which are turned on. The activation status may change as a device carrier moves. As discussed above, the information encoded in episodic data is much smaller than in continuous movement data. Many of the existing data analysis and privacy preservation methods designed for dealing with movement data are explicitly or implicitly based on the assumption of continuous objects movement between the measured positions and are therefore not suitable for episodic data. However, due to the increased availability of mobile phone data, analysis methods for episodic movement data and the retrieval of data for unobserved locations are rapidly evolving. Though such techniques pose a privacy risk, they also help us understand what sensitive information can be extracted from location traces.

Recent work, in [PTDC17] analyses which information you may gather from aggregated location data. They find that aggregates do leak information about punctual locations and profiles. They also highlight that the density of the observations, as well as their timing, play important roles. Their finding is that regular patterns in peak hours are better protected than sporadic movements. They also test whether data perturbation (compare Section 4) may provide additional protection. Their study reveals that perturbation in combination with aggregation offers little additional protection unless they introduce large amounts of noise that ultimately destroys the utility of the data.

More Information

T. Liebig. Smart navigation - chances, risk and challenges. In M. Jankowska, M. Pawelczyk, S. Augustyn, and M. Kulawiak, editors, *Navigation and Earth Observation - Law & Technology*, page (in press). IUS PUBLICUM, Warsaw, 2017

N. Andrienko, G. Andrienko, H. Stange, T. Liebig, and D. Hecker. Visual analytics for understanding spatial situations from episodic movement data. *KI - Künstliche Intelligenz*, pages 241–251, 2012

This publications was published in preparation for the VaVeL project and has no VaVeL acknowledgement.

3.2 Distributed Learning from Spatio-Temporal Aggregated Data

In case of distributed wireless sensors, a method for in-network learning is proposed in [LSM15, SLM15] (for the centralised case, compare e.g. [LXM13]). Our approach sends space-time aggregated values that, by design, provide k-anonymity. Hence, our method is privacy preserving and can be applied for large-scale traffic management scenarios. Our particular focus is on the prediction of future traffic flow at junctions throughout the region of interest (e.g. a city, a state or even areas at European scale).

Possible applications comprise, for instance,

- distributed car-to-car scenarios where cars or trucks communicate at junctions the number of observed vehicles at the road to estimate traffic flow and alter their individual transportation plans based on predicted traffic conditions, or,

- large scale traffic flow prediction that processes massive local observations on a high performance computer.

Scalable in-network algorithms belong to the field of distributed data mining. Existing work mostly focuses on horizontally partitioned data. There, full observations, i.e. all features and labels, are stored on different nodes in a network. However, network states representing the current traffic flow are *vertically partitioned*. Here, only partial information about observations is stored on different nodes. Learning and prediction therefore either require the transmission of observations or labels to other nodes. Previous work [DBV11, LSM12, SBDM13] has focused on sending less information about observations to a central coordinator. Here, we deal with reducing the amount of labels sent to neighboring peer nodes. Communication-efficient algorithms for vertical distributed learning are not just relevant for traffic flow prediction, but for applications as diverse as intrusion detection, monitoring production processes or smart grid management.

The task of learning from aggregated label information was first introduced in [KdF05]. Theoretical bounds have only recently been proven in [YKJC14]. [MCO07] propose variants of existing algorithms. The SVM optimization problem has been adapted to the setting [Rüp10, YLK+13]. Mean Map [QSCL09] estimates the mean operator solving a system of linear equations, while [PNCR14] extend it with a manifold regularization, outperforming both SVMs and Mean Map on standard datasets. A modified Kernel k-Means algorithm [CLQZ09] minimizes the distance to the given label proportions by matrix factorization. Recent work learns Bayesian network [HGIL13] and generative [FZY+14] classifiers. The LLP algorithm proposed in [SM11] first determines clusters and then tries to label them. LLP only has linear running time, while its prediction performance competes with the approaches in [QSCL09, Rüp10] and [CLQZ09].

Given are m distributed sensor nodes $P_1, \ldots, P_m$. Each sensor node P_i delivers an infinite series of real-valued measurements $\ldots, v_{t-1}^{(i)}, v_t^{(i)}, v_{t+1}^{(i)}, \ldots$ for different time points $\ldots, t-1, t, t+1, \ldots$. Time spans between two measurements are equidistant, given a constant sample rate. Let t denote the current time of measurement, while $t-a$ and $t+a$ are time points a steps in the past and future. Each sensor node also has a spatial location.

The task, given the current time point t, is therefore to predict a label y from a set $Y = \{Y_1, \ldots, Y_l\}$ of distinct categories at some arbitrary node P_i at future time point $t+r$, based on the current and previous (raw) sensor readings at all or a subset of nodes $P_1, \ldots, P_m$.

We assume that for learning, measurements and labels are somehow recorded (see below) over a fixed-length time period. For the supervised training of prediction models, each node P_i thus provides a sequence $V_i = \langle v_1^{(i)}, \ldots, v_n^{(i)} \rangle$ of measurements, $v_j^{(i)} \in \mathbb{R}$, and a sequence $L_i = \langle y_1^{(i)}, \ldots, y_n^{(i)} \rangle$ of labels $y_j^{(i)} \in Y$.

Instead of centralizing all data, we propose that each P_i records and stores its own measurements and labels. For predicting future traffic flow categories at node P_i, we restrict learning to P_i itself and c topological neighboring nodes around P_i. For instance, to learn and predict the future type of traffic flow at some street junction, considered are only measurements and labels recorded at the junction itself and at c junctions closest to it.

Before training, each P_i preprocesses measurements V_i as follows. A window of size p is slided over the series V_i with step size 1, storing all thereby created windows $x_t^{(i)} = \{v_{t-p+1}^{(i)}, \ldots, v_t^{(i)}\}$, $t = p, \ldots, n$ as rows in a dataset D_i. Let $N^{(i)} = \{n_1^{(i)}, \ldots, n_c^{(i)}\}$ be the set of indices for the

c neighboring nodes around P_i. Based on the datasets $D_i, D_{n_1^{(i)}}, \ldots, D_{n_c^{(i)}}$ and labels L_i, we want to learn a *local* function (model) $f^{(i)}$ that, given windows $x_t^{(i)}, x_t^{n_1^{(i)}}, \ldots, x_t^{n_c^{(i)}}$ of sensor readings from node P_i and its neighbors, predicts the label $y_{t+r}^{(i)}$ at node P_i with horizon r correctly.

Interpreting windows $x_t^{(i)}, x_t^{n_1^{(i)}}, \ldots, x_t^{n_c^{(i)}}$ as features of a single observation x that should be classified, the data is *vertically partitioned*, since each neighboring node of P_i only stores partial information about x, i.e. a subset of features.

An obvious choice for the training of $f^{(i)}$ at P_i is to ask for the recorded measurements at each neighboring node, concatenate their columns at P_i and join the labels stored at P_i to the new dataset. The approach is more scalable than centralizing all data, since the number c of neighbors is fixed, avoiding the bottleneck problem of limited bandwidth. However, each node still needs to transmit *all* measurements to each of its neighbors, consuming at least as much energy per node as sending all data to a single server.

Therefore, we propose to send only label information from node P_i to its neighbors and to train models $f_0^{(i)}$ at node P_i and $f_{n_1^{(i)}}^{(i)}, \ldots, f_{n_c^{(i)}}^{(i)}$ at its neighbors. As model $f^{(i)}$ at node P_i, we propose a majority vote over predictions from itself and its neighboring nodes. All models are *local*, since they only consider measurements and labels of a fixed number of close topological neighboring nodes around P_i. Moreover, the approach works fully *in-network* without a central coordinator, since each node only communicates with its neighboring peer nodes. As learners at each node, one may consider supervised learners, like kNN, Decision Trees or SVMs. Considering the limited computational resources of sensor nodes, however, our evaluation is solely based on kNN.

Since the number of bits to encode all labels is often less than an encoding of all measurements, communication is saved by sending labels *from* P_i instead of measurements *to* P_i. However, supervised learning still requires individual labels for all observations. The question is if communication can be reduced even further, by sending fewer labels or aggregated label information to each neighboring node.

Semi-supervised [CSZ06] and active learning [BHV10] show that training on fewer labels may achieve a similar performance as training on all labels. However, such methods do not preserve the privacy of the data, since they need individual labels of observations (see Sect. 3.2). Instead, we propose to send only *aggregated* label information, i.e. label counts, to neighboring nodes for learning.

Before sending label information to each of its neighboring nodes, P_i divides its time-related sequence L_i of labels into consecutive batches $C_1^{(i)}, \ldots, C_h^{(i)}$ of a fixed size b. It respects the prediction horizon r, such that each $C_j^{(i)}$ consists of labels from time point $t + (j-1)b + r$ to $t + jb + r$ and align correctly with time points of observations (i.e. windows of measurements) at other nodes. Let n be from here on the size of datasets D_i, i.e. the number of windows stored. Then, h is $\lceil n/bs \rceil$. For each batch j, labels $y \in Y$ are aggregated by counting how often they occur, and stored in a $h \times l$ matrix of label counts $Q^{(i)} = (q_{jd}^{(i)})$, where $q_{jd}^{(i)} = |\{y \in C_j | y = Y_d\}|$.

Let $P_{n_e^{(i)}}$ be a neighboring node receiving label counts from P_i. $P_{n_e^{(i)}}$ transforms $Q^{(i)}$ into a label proportion matrix $\Pi^{(i)} = (\pi_{jd}^{(i)}) = q_{jd}^{(i)}/b$, i.e. the counts of labels are divided by batch

size b. Since every node knows b and r, $P_{n_e^{(i)}}$ can partition its own windows $x_1^{n_e^{(i)}}, \ldots, x_n^{n_e^{(i)}}$ of measurements into batches $B_1^{n_e^{(i)}}, \ldots, B_h^{n_e^{(i)}}$. Since the sender respects r, the time spans used for aggregating the labels align correctly with the windows of measurements stored at $P_{n_e^{(i)}}$.

The learning task at node $P_{n_e^{(i)}}$ now consists of learning a model $f_{n_e^{(i)}}^{(i)}$, only based on its batches of (unlabeled) measurements and the label information from node P_i, stored in the label proportion matrix $\Pi^{(i)}$, such that the expected prediction error over individual observations is minimized. This task is also known as *learning from label proportions*.

Several methods have been developed to solve the task. Considering the limited computational resources of sensor nodes, the LLP algorithm [SM11] looked most promising, since LLP has a linear running time and its centroid model a small memory footprint. Moreover, it can handle multi-class classification problems as they arise in traffic monitoring. However, we found that it still needs to be improved for scalability issues and performance.

LLP learns from label proportions by first clustering all observations and then assigning labels to each cluster. The task of cluster analysis consists of partitioning a set of observations into a set $\mathcal{C}$ of k disjunct groups (clusters) $C_1, \ldots, C_k$, such that the similarity of observations in each cluster is minimized. LLP relies on the idea that observations having the same class also share similar features, i.e. that clusters somehow correspond to classes. LLP allows for several clusters per class and assumes that the majority of elements of a cluster belongs to the same class. Once given a clustering the only remaining problem is to assign correct labels to each cluster.

More formally, let $\mu : X \to \mathcal{C}$ be a mapping that assigns an arbitrary observation $x \in X$ to a cluster $C \in \mathcal{C}$. For centroids $c_1, \ldots, c_k$ found with k-Means, $\mu(x)$ would be defined as $\mu(x) = \text{argmin}_{C_k \in \mathcal{C}} \, ||x - c_k||^2$.

Further, let $\ell : \mathcal{C} \to Y$ be a mapping which assigns a label $\lambda \in Y$ to each cluster $C \in \mathcal{C}$. For ease of notation, let f denote model $f_{n_e^{(i)}}^{(i)}$ to be learned at node $P_{n_e^{(i)}}$, B_i denote the batch $B_i^{n_e^{(i)}}$ and Π denote matrix $\Pi^{(i)}$. f is the composition of mappings ℓ and μ, i.e. $f = \ell \circ \mu$.

With prediction model f, entries γ_{jd} of a model-based proportion matrix $\Gamma_f = (\gamma_{jd})$ can be calculated as

$$\gamma_{jd} = \frac{1}{|B_j|} \sum_{x \in B_j} I(f(x), Y_d), \quad I = \begin{cases} 1 & : \quad f(x) = Y_d \\ 0 & : \quad f(x) \neq Y_d \end{cases} . \tag{1}$$

The LLP algorithm now minimizes the mean squared error

$$\text{MSE}(\Pi, \Gamma_f) = \frac{1}{hl} \sum_{j=1}^{h} \sum_{d=1}^{l} (\pi_{jd} - \gamma_{jd})^2 \, , \tag{2}$$

between the given label proportion matrix Π and the model-based proportion matrix Γ_f by trying different label mappings ℓ.

A Local Search Strategy with Multistarts

The LLP algorithm as introduced in [SM11] can work with different cluster algorithms and labeling strategies. LLP with an exhaustive labeling strategy, called LLP_{exh} in the following,

tries all possible labellings of the clusters. We found it too time-consuming for the evaluations at the end of the section. The greedy strategy, initially proposed in [SM11], didn't achieve sufficient accuracies for traffic prediction. Hence, a better search strategy is demanded.

We propose a local search that is started multiple times with different random combinations of labels. LLP with this search strategy will be called LLP_{lsm} in the following. The local search greedily improves on the current labeling of clusters by trying all possible labels at each component of a labeling vector λ. Fitness measures how well the model-based label proportion matrix Γ_f, as calculated from the current labeling, matches the given label proportions. If the fitness improves, the search starts from the first component of the labeling vector λ, again. Otherwise, it resets the label at the current position $kpos$ to the label of the best (local) solution found so far. The returned value is the best labeling found over all starts of the different greedy searches.

In each iteration, the greedy search runs until no further improvement is possible. Moreover, at each step of the algorithm, the fitness either improves or is staying the same (which is a stopping criterion). Therefore, each search finds a local minimum. Since the number of searches is finite, the returned labeling vector is also locally minimal. In comparison to LLP_{exh}, it cannot be guaranteed that a globally optimal solution is found. However, with regard to the prediction results presented at the end of this section, we found that a local search performed sufficient enough, despite a much lower running time.

LLP as introduced in [SM11] combines the MSE with two other error measures. However, we found that the use of these additional measures decreases the accuracy in the traffic monitoring scenario. Hence, all experiments inthe following evaluation are based on the MSE, only. Similarly, we abstain from the evolutionary feature weighting presented in [SM11], since it would heavily increase the algorithm's running time.

Analysis of Communication Costs

Each node P_i transmits a matrix Q to each of its neighboring nodes, consisting of counts for each label $Y_d \in Y$ and batch. Such counts may be assumed to be integers. The maximum value of each integer is b, which means we need to reserve at most $\lceil \log_2 b \rceil$ bits for each label. The number of batches, given n observations, is $\lceil n/b \rceil$. The total number of bits z_{AGG} for encoding matrix Q is therefore

$$z_{\text{AGG}} = \left\lceil \frac{n}{b} \right\rceil \lceil \log_2 b \rceil |Y| \quad . \tag{3}$$

In comparison, the number of bits z_{ALL} required to encode *all* labels of n observations, for $|Y|$ different labels, is at most

$$z_{\text{ALL}} = n \lceil \log_2 |Y| \rceil \quad . \tag{4}$$

The total costs are then either z_{AGG} or z_{ALL}, multiplied by the number of nodes m. Here we assume that label information is broadcast to each neighboring node, which is not unrealistic for sensors in topologically close regions. All payloads reported in this section base on this assumption.

Analysis of Privacy

The *vulnerable data* are the original sensor readings. These traffic flow measurements bare the risk of re-identification of individual vehicles. For example in a dense sensor network with sparse observations of vehicles, their occurrence may be tracked throughout the network. As mobility often is a regular behaviour and contains patterns this risk is even higher. In this section we show that our LLP_{lsm}-based algorithm transforms the data such that re-identification risk is at most $1/s$.

In our distributed setting, *adversaries* of a particular sensor node are malicious sensors that could use received measurements of neighboring sensors for deduction of individual mobility traces. The following *attack model* is possible: The adversary analyses differences among neighboring sensor readings and deduces individual movement. If the difference among two neighboring sensor readings is zero and both traffic flow counts are w, it is (depending on network topology) likely that w vehicles moved between the two sensors. In case of three neighboring sensors P_a, P_b, P_c their measurements v_a, v_b, v_c can be combined as follows: If $v_a - v_b = w = v_c$ it may be deduced that on the way from P_a to P_b w vehicles turned to P_c, in case $v_a - v_b = -w = -v_c$ w vehicles originated from the location P_a.

With our new LLP_{lsm}-based approach we process discretized traffic flow values and just communicate counts of these value ranges. We denote the minimal (nonzero) interval width by s. Thus, measurements may not be distinguished up to a granularity of s vehicles and w is bounded by s, $w \geq s$. In turn, the risk of re-identification with the hereby described attack model is at most $1/s$. Our approach therefore provides s-anonymity by design. The aggregation of label information reduces the remaining risk for disclosure of neighboring labels at a malicious sensor node. The solely transmission of label counts prevents doubtless reconstruction of the labels [YKJC14].

We perform tests of the method on data of the city of Dublin. The Sydney Coordinated Adaptive Traffic System (SCATS) provides information on vehicular traffic at over 750 fixed sensor locations as spatio-temporal time series [McC14]. The data we use[1] is a snapshot from 01/01/2013 till 14/05/2013, consisting of tuples (t, u, w), where u is the location of the observation and consists of an index for the junction, the arm and the lane number at which the sensor is located at. The metric w contains the aggregated vehicle count at sensor location since last measurement. The time stamp t denotes the recording time.

3.3 Experimental Results

Instead of using simple toy experiments, based on simulation (e.g. [LKW12]), we directly apply our method to the dataset from the SCATS sensors at Dublin. Local models are trained for each of the 296 sensor nodes and their nearest topological neighbors. As supervised base-line learner that receives all labels, we use kNN with $k = 15$. For learning from aggregated label counts, we cluster the observations at each node with k-Means ($k = 15$, 50 different random starting points, 500 iterations at maximum) and label the clusters with LLP_{lsm} (with 150 starts of the local greedy search) at each node for different batch sizes $b = 25, 50, 75$ and 100. The accuracy of each method is assessed by a 10-fold cross validation, i.e. all models are trained and evaluated for different hold-out sets 10 times. In total $296 \times 7 \times 10 = 20,720$ models for

[1] Data is publicly available at http://dublinked.ie .

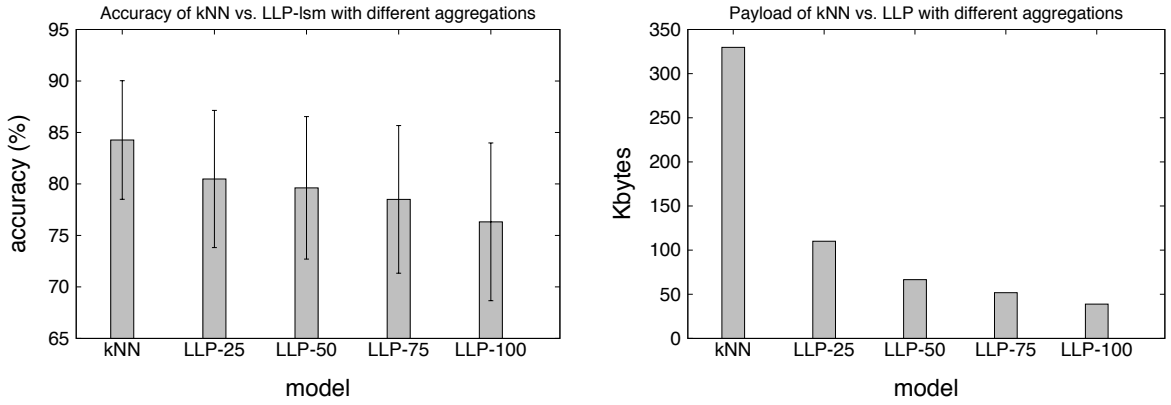


Figure 2: Trade-off between accuracy and payload sent for kNN and LLP_{lsm}

kNN need to be evaluated and $296 \times 7 \times 10 \times 4 = 82,880$ models trained and evaluated for LLP_{lsm}. The evaluation has been done offline in parallel on different machines (about 36 CPU cores).

Figure 2 shows the trade-off between accuracy and payload sent for kNN and LLP_{lsm} trained on differently sized batches of aggregated labels. Besides the average accuracy over all 10-fold cross-validations at each node, the bars in Fig. 2 (left) also depict the standard deviation of accuracy over all nodes. In general, LLP_{lsm} performs slightly worse than kNN. Nevertheless, there are still many junctions for which the traffic flow is predicted quite well with LLP_{lsm}. Some locations have bad performance with both methods, a comparison to the map reveals that these are locations of parking areas e.g. inner-city parking houses and recreational areas where many vehicles stay for a long period of time.

More Information

T. Liebig, M. Stolpe, and K. Morik. Distributed traffic flow prediction with label proportions: From in-network towards high performance computation with mpi. In G. Andrienko, D. Gunopulos, I. Katakis, T. Liebig, S. Mannor, K. Morik, and F. Schnitzler, editors, *Proceedings of the 2nd International Workshop on Mining Urban Data (MUD2)*, volume 1392 of *CEUR Workshop Proceedings*, pages 36–43. CEUR-WS, 2015

M. Stolpe, T. Liebig, and K. Morik. Communication-efficient learning of traffic flow in a network of wireless presence sensors. In *Proceedings of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases (PDCKDD 2015)*, CEUR Workshop Proceedings. CEUR-WS, 2015

Both publications were published in preparation for the VaVeL project and have no VaVeL acknowledgement.

3.4 Centralized Learning from Spatio-Temporal Aggregated Data

If aggregated data is centralized, it is important that the central authority can not reveal individual measurements as they could help to (re-)identify individual persons (compare [SCR+11]). Consider, e.g., stationary sensors that monitor the number of vehicles passing over time. Even aggregated counts over time intervals may reveal individual mobility pattern, if combined with neighbouring observations. Our approach to protect privacy in these cases is to use homeomorphic encryption, see Section 6. Also another recent work [PDCR16] uses homeomorphic encryption in combination with Succint Sketches [MDDC15] (compare Section 5) to analyze aggregated data at a centralized authority.

More Information

T. Liebig. Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services*, pages 86–99, 2014

T. Liebig. Privacy preserving centralized counting of moving objects. In F. Bacao, M. Y. Santos, and M. Painho, editors, *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, pages 91–103. Springer International Publishing, 2015

Both publications were published in preparation for the VaVeL project and have no VaVeL acknowledgement.

4 Privacy via Data Perturbation

In contrast to data aggregation, data perturbation addresses a group of methods that aim to remove vulnerable data by randomization, splitting, deletion or simplification. Most prominent concept of this group is differential privacy, we will provide a brief overview in next subsection and continue with the other topics. A contribution of the VaVeL consortium is in Section 4.4 which briefly reflects the work on filtering vulnerable data items in a crowdsourcing application (compare also Deliverable 4.2 Section 4.3).

4.1 Differential Privacy

In [CY16] a privacy scheme is presented that transforms the original data such that the data is as similar as possible to the original one and the individual privacy is ensured. The application scenario, their work focuses on, is counting the number of people within a region of interest. A de-facto standard for privacy preserving data publishing is ϵ-Differential Privacy (ϵ-DP). A randomized algorithm A achieves ϵ-DP, if it satisfies

$$\frac{P(A(Q(D)))}{P(A(Q(D^*)))} \leq e^{\epsilon}, \epsilon > 0$$

D^* denotes the database except the individual data [DMNS06a], ϵ may be seen as 'privacy budget' – an unitary privacy level control. The differential privacy approach is robust under

linkage attack. Randomization of the algorithm is ensured by application of noise. Literature proposes two distributions for that noise a) Laplacian [DMNS06a] and b) Exponential [MT07] distribution. Differential privacy protects the database against a predefined number of queries, in case of an infinite data stream and continuous access to statistics over this stream it is not sufficient to add some noise to individual data items, as the parameters of the noise may easily be disclosed.

Therefore, [CY16] proposes a method to achieve l-trajectory privacy. Similar to the k in k-Anonymity, their l denotes the number of indistinguishable trajectories. The application the authors aim for is publishing of statistics (i.e. counts) over trajectories. Their approach consists of two elements:

- a Greedy Algorithm (GA) that adds Laplacian noise to the data elements in each time step,
- a re-publish strategy that publishes the data with minimum Manhattan distance to the original data.

The GA selects for every time stamp t in a time window $\epsilon(t)$ such that following $\epsilon(t+1)$ can by maximal, while keeping the sum over all $\epsilon(t)$ less than ϵ. This algorithm introduces too much noise. Thus, the Minimum Manhattan Distance Mechanism was proposed in [MT07] to improve utility of the published counts.

4.2 Randomisation

The work in [Sie16] focuses on sparse transaction data, which is one possible representation of trajectories. In case of a characteristic function for a set of street segments in a city, one may transform any trajectory in this city to a binary vector that holds a 1 for every segment within the trajectory and a 0 otherwise (compare [LKM08, LKM09]). The authors define `unicity`, a probability that knowing p transactions of a user is enough to identify the complete trail of that user. The problem is formulated in terms of support of patterns in a database:

- Small support pattern should have mostly random support,
- while large(r) support patterns should mostly keep their original support.

The method they propose is randomization. Therefore, they encoded the database using a code table CT. A code table is a two column table in the first column there are patterns in the second column there are code words from some prefix code. The code table with the minimum description length (MDL) is optimal for database generation, it minimizes $L(H) + L(D|H)$, where $L(\cdot)$ is the number of bits required for describing H the code table, and $D|H$ the data when encoded with H. The algorithms KRIMP [VVLS11] and SHRIMP [HPM14] are known to provide good approximations. After the compression of the database in a code table, this code table is used in [Sie16] as a generative model to sample a database that provides required properties.

The method, however, is applicable to transaction databases and has problems when time is also considered. In this case sequences have to be considered. The work in [MPPP14] proposes privacy measures on sequence datasets. Similarly to [Sie16], [MPPP14] describes the construction of a k-anonymous version of a sequence dataset.

4.3 Splitting and Deletion

Another group of methods focuses on the privacy preserving publication and centralization of trajectory data. As seen in the motivation section (Section 2) this causes multiple (re-)identification risks. The group of methods addressed in this section splits these trajectories in non-identifying pieces or even deletes parts to ensure privacy.

The recent work in [TPMS16] suppresses the locations and splits trajectories to prevent an attacker from utilizing partial knowledge on visited location to infer unknown visits. Their method is applied to episodic movement data (as for example recorded by RFID technology), compare Section 3.1.

The work in [WBCL16] aims at usage in mobile sensors, and detects privacy relevant locations for the individuals, these are clustered and whenever the person stays at these locations its position is cloaked and no data is sent to the global authority.

4.4 Filtering

The work presented in [BK16] (and discussed in detail in D4.2 Section 4.3) focuses on private sharing trajectory data of persons and the problem that an adversary may link the points even if personal identifiers are not given (see Section 2.2.3). The method uses clustering to identify the likelihood of re-identification of one data tuple containing position and time. The entropy of the data points is used to identify crowded and uncrowded points, such that the method automatically detects vulnerable data tuples and prevents sharing.

More Information

Deliverable 4.2 - Section 4.3 and

I. Boutsis and V. Kalogeraki. Location privacy for crowdsourcing applications. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 694–705, New York, NY, USA, 2016. ACM

4.5 Simplification and Generalization

Another method to guarantee privacy is to modify the trajectories in database by generalization, sometimes also called simplification ([SL16, WSN16, GFTB17]). This group of methods identifies common stay points in the trajectories and common movement directions and reduces the trajectory information to these common substructures. The trade-off utility vs privacy is then controlled by a parameter that adjusts the sensitivity of the merge operations.

5 Privacy via Sketches

In order to monitor streams, sketches and summaries became a common tool to store memory constrained information on a data stream in a timely manner. For an introduction to sketches, we refer the interested reader to [CMY08]. Very prominent examples are the Count-Min Sketch

[CM05] and Count Linear Flajolet-Martin (FM) Sketch [FFGM07]. A Count-Min sketch is a two-dimensional table with a width w and a depth d. Each entry in the table is initialized with zero. A set of d hash functions are chosen uniformly at random mapping $\{0,1\}^* \to \{0,1\}^w$. To update item i by a quantity c_i in every row $j \in [1,\ldots,d]$, we apply h_j to identify which bucket to increase by c_i. Using this sketch, the count of item i may then be estimated by retrieving the minimum of the estimates of $c_i = X[j, h_j(i)]$ from every row $j \in [1,\ldots,d]$. The retrieved value is an upper bound, its approximation becomes the more accurate the more buckets d are used. The memory usage of these sketches may become very high and the algorithm is not applicable if you want to monitor a large number of elements. FM sketches overcome this limitation as their space requirement is logarithmic in the number of monitored objects [FFGM07]. In [KNW10] the memory consumption is slightly higher, but their sketching approach provides a linear update time, and is thus optimal.

The analysis of privacy using these FM sketches is performed in [KKM+13]. In their work, the authors focus on two scenarios using stationary wireless sensors: 1) crowd monitoring and 2) flow monitoring. Whilst in the first one the covered regions of the sensors overlap, they are mostly separated in the second scenario. For mobility monitoring it is important to re-identify a person amongst various locations, on the other hand the individual mobility patterns should be protected. In their work, they use FM sketches (per mote in the sensor network) to count the number of distinct moving objects each sensor monitored. In their work they highlighted that it is possible to combine the sketches amongst multiple sensors and provide probabilistic bounds on the observed numbers visiting and co-visiting the places.

Another recent work in [MDDC15] applies Count-Min sketches in combination with homeomorphic encryption, compare Section 6. In a distributed scenario every party computes sketches over the part of the stream it monitors. Given some asymmetric encryption protocol with homeomorphic properties (compare Section 6), these values are encrypted and sent to a central authority. The central authority aggregates the encrypted sketches and decrypts the result. The resulting decrypted aggregate sketches are then communicated to each party that may utilize it for k-NN queries.

6 Privacy via Homeomorphic Encryption

An interesting method to protect vulnerable data is the usage of homeomorphic encryption. The methods have in common that vulnerable data is encrypted, analysis is performed on the cryptotext and its decryption reveals the result of the analysis without directly accessing the underlying data items. In [GLN13] the authors show that with this homeomorphic encryption scheme also more difficult algorithms and machine learning can be performed. Three utilized encryption schemes can be categorized in three different concepts that provide varying utility on the cryptotext:

- additive homeomorphic encryption,
- order preserving homeomorphic encryption,
- fully homeomorphic encryption,

In the following, we briefly discuss methods for data analysis using these encryption techniques.

6.1 Additive Homeomorphic Encryption Scheme

The Paillier Cryptosystem [Pai99] is an additive homeomorphic encryption scheme that allows for secured addition of plaintext messages by multiplication of their cryptotext. Therefore, asymmetric encryption is utilized and the encryption scheme consists of the three components: (1) key generation, (2) encryption, and (3) decryption. In the following, we give a brief introduction, but we also recommend the interested reader to follow these publications [O'K08, Ste10]. The method bases on asymmetric encryption schemes (e.g. RSA method [RSA83]) that use a public key to encode a message and a private key to decode it again. The RSA method uses one-way functions. These are functions which are easy to compute in one direction but difficult to reverse. A simple metaphor of this function is a phone book: While it is easy to derive the call number of a particular person, it is hard to look up the name given a phone number. Preliminary for understanding is the notion of multiplicative inverse b of a number a, which is defined as $a \cdot b = 1 \text{ mod } m$. This inverse only exists, if m and a are co-prime, i.e. $gcd(m, a) = 1$.

For a better understanding let us first consider a RSA encrypted communication among a client who wants to send a message to a server. In this case, the system works as follows. In a key generation process, the server chooses two different primes p and q and computes $n = pq$ and $m = (p-1)(q-1)$. Furthermore, the server chooses a number a which is co-prime to m. The public key, created by the server, then denotes as $pk = (n, a)$. The server computes the multiplicative inverse $b = a^{-1} \text{ mod } m$ of a, which is the secret private key.

Encryption:
The client has a message x, with $x < m$. He sends the ciphertext c, computed as

$$E(x, pk) = x^a \text{ mod } n \ . \tag{5}$$

Decryption:
The server decrypts the message and restores the plaintext by computing

$$x = D(c) = c^b \text{ mod } n \ . \tag{6}$$

The system is secure, as knowledge of n does not reveal p and q, since factorization is in NP [Joh84].

A public key encryption scheme (E, D), where E and D are algorithms for encryption and decryption, is homomorphic when it meets the condition $D(E(m_1) \cdot E(m_2)) = m_1 + m_2$

Our approach bases on the generalisation of Paillier's public-key system [Pai99], introduced in [DJ01]. Their crypto system uses computations modulo n^{s+1}, with n being the RSA modulus and s a natural number. By setting $s = 1$ Paillier's scheme is a special case [Pai99]. If $n = pq$ with p and q being odd primes, then the multiplicative group $\mathbb{Z}^*_{n^{s+1}}$ is a direct product of $G \times H$, where G is of cyclic order n^s and H is isomorphic to $\mathbb{Z}^*_n$. Thus, $\bar{G} = \mathbb{Z}^*_{n^{s+1}}/H$ is cyclic of order n^s.

For an arbitrary element $a \in \mathbb{Z}^*_{n^{s+1}}$ $\bar{a} = aH$ denotes the element represented by a in the factor group $\bar{G}$. Thus, we choose a $g \in \mathbb{Z}^*_{n^{s+1}}$ such that $g = (1+n)^j x \text{ mod } n^{s+1}$ for known j relatively prime to n and $x \in H$. Let λ be the least common multiplier of $p-1$ and $p-1$, $\lambda := lcm(p-1, q-1)$.

Key Generation in Paillier's crypto system is as follows: Choose two primes p and q, and $n = p \cdot q$, g with $|g|$ is n in $\mathbb{Z}^*_{n^2}$. the public key then is the tuple (n, g).

Carmichael's function $\lambda(n)$ is the smallest m for an integer n, such that $\forall a, gcd(a,n) = 1$: $a^m \equiv 1 \text{ mod } n$.

If p and q are prime factors of n, $\lambda(n) = lcm(\phi(p), \phi(q))$. Thus, in case, p and q are prime, holds $\lambda(n) = lcm(p-1, q-1)$. λ then is the private key.

In a exemplified implementation in Cran-R this equals to

```
require("pracma")   # provides gcd()
require("numbers")  # provides isPrime(), modpower()

p=3
q=5

# begin key generation
n=p*q
phi=(p-1)*(q-1)
g=n+1
lambda=Lcm((p-1),(q-1)) # = (p-1)*(q-1)/gcd((p-1),(q-1))
```

In this example, the resulting values should be $n = 15$, $\phi(n) = 8$, $\lambda(n) = 4$, $g = 16$

Properties As stated above, after generating these keys, following properties hold. $\forall w \in \mathbb{Z}^*_{n^2}$: $w^\lambda \equiv 1 \text{ mod } n$ and $w^{n\lambda} \equiv 1 \text{ mod } n^2$. The set $\mathcal{S}_n = \{u < n^2 | u = 1 \text{ mod } n\}$ is a multiplicative subset modulo n^2 for which the function $L(u) = (u-1)/n$ is well-defined for all $u \in \mathcal{S}_n$.

Encryption of the plaintext m, requires that m is element of $\mathbb{Z}_{n^s}$. Then, we choose at random $r \in \mathbb{Z}^*_{n^{s+1}}$. The ciphertext $E(m,r)$ computes as:

$$\begin{aligned} \mathcal{E}_g : \mathbb{Z}_n \times \mathbb{Z}^*_n &\rightarrow \mathbb{Z}^*_{n^2} \\ \mathcal{E}_g(x,y) &\rightarrow g^x \cdot y^n \text{ mod } n^2 \\ \mathcal{E}_g(m,r) &\rightarrow g^m \cdot r^n \text{ mod } n^2 \end{aligned}$$

The corresponding code snippet could be similar to:

```
encrypt <- function(m,n) {
  r=n
  while (gcd(r,n)!=1) {
    r=sample(1:n,1)
  }
  c=(modpower(g,m,n^2)*modpower(r,n,n^2)) %% (n^2)
  return(c)
}
```

Decryption For the ciphertext c compute c^d mod n^{s+1}. If $c = E(m,r)$ this results in

$$\begin{aligned} c^d &= (g^m r^{n^s})^d = E(m,r) \\ &= ((1+n)^{jm} x^i r^{n^s})^d \\ &= (1+n)^{jmd \text{ mod } n^s} (x^m r^{n^s})^{d \text{ mod } \lambda} \\ &= (1+n)^{jmd \text{ mod } n^s} . \end{aligned} \tag{7}$$

In [DJ01] an algorithm is proposed to compute jmd mod n^s. Their method bases on a function $L(b) = (b-1)/n$ which ensures that

$$L((1+n)^i \text{ mod } n^{s+1}) = (i + \binom{i}{2} n + \ldots + \binom{i}{s} n^{s+1}) \text{ mod } n^s \; . \tag{8}$$

The basic idea of their algorithm is to compute the value iteratively in a loop by increasing s, as $L(1+n)^i$ mod $n^2 = i$ mod n. With the same method computed for g instead of c the value jd mod n^s is computed. The plaintext then is:

$$m = L(c^\lambda \text{ mod } n^2) * L^{-1}(g^\lambda \text{ mod } n^2)$$

with $L(x) = (x-1)/n$

To follow these steps, it is useful to inspect the involved components $A := \mathbb{Z}^*_{n^2}$,A^λ,$L(A\lambda)$ mod n^s, $L(A^\lambda)$ mod n^2 and with some exemplified p and q. Next, we resume the implementation we started above and inspect these values.

computation of $A = \mathbb{Z}^*_{n^2}$

```
A=matrix(ncol=n,nrow=phi)
j=0
for (i in 1:phi) {
 j=j+1
 while (gcd(j,n)!=1) {j=j+1}
 for (m in 0:(n-1)) {
  A[i,(m+1)]=(modpower(g,m,(n^2)) * modpower(j,n,(n^2))) %% n^2
 }
}
```

Inspection of A

```
>print(A)

 r\m    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
 1      1   16   31   46   61   76   91  106  121  136  151  166  181  196  211
 2    143   38  158   53  173   68  188   83  203   98  218  113    8  128   23
 4    199   34   94  154  214   49  109  169    4   64  124  184   19   79  139
 7    118   88   58   28  223  193  163  133  103   73   43   13  208  178  148
 8    107  137  167  197    2   32   62   92  122  152  182  212   17   47   77
11     26  191  131   71   11  176  116   56  221  161  101   41  206  146   86
13     82  187   67  172   52  157   37  142   22  127    7  112  217   97  202
14    224  209  194  179  164  149  134  119  104   89   74   59   44   29   14
```

Inspection of A^λ

```
>modpower(A,lambda,n^2)

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[2,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[3,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
```

```
[4,]    1   61  121  181   16   76  136  196   31   91  151  211   46  106  166
[5,]    1   61  121  181   16   76  136  196   31   91  151  211   46  106  166
[6,]    1   61  121  181   16   76  136  196   31   91  151  211   46  106  166
[7,]    1   61  121  181   16   76  136  196   31   91  151  211   46  106  166
[8,]    1   61  121  181   16   76  136  196   31   91  151  211   46  106  166
```

Inspection of $L(A^\lambda)$ mod n^2

```
>((modpower(A,lambda,n^2)-1)/n)

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[2,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[3,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[4,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[5,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[6,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[7,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[8,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
```

Inspection of $L(A^\lambda)/L(g^\lambda)$ mod n
$L(g^\lambda \text{ mod } n^2) = L(61) = 60/15 = 4$
$L^{-1}(g^\lambda \text{ mod } n^2) = 4$

```
> Lg = (modpower(g,lambda,n^2)-1)/n
> Lg_inv = (extGCD(Lg,n)[2]+n) %% n
>(((modpower(A,lambda,n^2)-1)/n)  * Lg_inv) %% n

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[2,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[3,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[4,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[5,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[6,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[7,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[8,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
```

As it can be seen in the last step, combination of previous transformations reveals the column coordinates of some particular message, this is then used for encryption and guarantees as well homeomorphic properties. In this example, it can easily be seen that $92 = (191 * 37) \text{ mod } 15^2$

Decryption Examples Here are some examples for manual decryption, first we denote some exemplified computation

$$(((((77^4) \text{ mod } (15^2)) - 1)/15) * 4) \text{ mod } 15 = 14$$

and then some Cran-R codes, for further tests

```
(((((7^4) %% (15^2))-1)/15) * 4) %% 15 = 10
(((((164^4) %% (15^2))-1)/15) * 4) %% 15 = 4
(((((82^4) %% (15^2))-1)/15) * 4) %% 15 = 0
(((((98^4) %% (15^2))-1)/15) * 4) %% 15 = 9
```

The Decrypt function may then be implemented as

```
decrypt <- function(c,lambda,n) {
  c=c %% n^2
  Lg = (modpower(g,lambda,n^2)-1)/n
  Lg_inv = (extGCD(Lg,n)[2]+n) %% n

  m = ((modpower(c,lambda,n^2)-1)/n * Lg_inv) %% n
  return(m)
}
```

6.1.1 Complete Sourcecode

```
require("pracma")   # provides gcd()
require("numbers")  # provides isPrime(), modpower()

p=3
q=5
n=p*q
phi=(p-1)*(q-1)
g=n+1
lambda=Lcm((p-1),(q-1)) # = (p-1)*(q-1)/gcd((p-1),(q-1))

encrypt <- function(m,n) {
  r=n
  while (gcd(r,n)!=1) {
    r=sample(1:n,1)
  }
  c=(modpower(g,m,n^2)*modpower(r,n,n^2)) %% (n^2)
  return(c)
}

decrypt <- function(c,lambda,n) {
  c=c %% n^2
  Lg = (modpower(g,lambda,n^2)-1)/n
  Lg_inv = (extGCD(Lg,n)[2]+n) %% n

  m = ((modpower(c,lambda,n^2)-1)/n * Lg_inv) %% n
  return(m)
}

# correctness
if (5 == decrypt(encrypt(5,n),lambda,n)) {
   print("m = D(E(m)) seems to be ok")
}

# Homeomorphism
```

```
if (5 == decrypt(encrypt(2,n)*encrypt(3,n),lambda,n)) {
   print("m1+m2 = D(E(m1)*E(m2)) seems to be ok")
}
```

More Information

T. Liebig. Das Paillier Cryptosystem mit Beispielen in Cran-R. http://www-ai.cs.uni-dortmund.de/LEHRE/FACHPROJEKT/WS1516/download/paillier.pdf, 2015. [Online; accessed 14-Dezember-2016]

T. Liebig. Privacy preserving centralized counting of moving objects. In F. Bacao, M. Y. Santos, and M. Painho, editors, *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, pages 91–103. Springer International Publishing, 2015

T. Liebig. Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services*, pages 86–99, 2014

The last two publications were published in preparation for VaVeL and have no VaVeL acknowledgement. The first one is course material for a Bachelor's course on privacy preserving mobility data analysis.

6.2 Privacy Preserving Centralized Counting of Moving Objects

In this section we present our work on privacy preserving aggregation framework for distributed data streams. Individual location data is obfuscated to the server and just aggregates of *k* citizens can be processed. This is ensured by use of Pailler's homomorphic encryption framework and Shamir's secret sharing procedure. In result we obtain anonymous unification of the data streams in an un-trusted environment.

6.2.1 Shamir's Secret Sharing

The work presented in [Sha79] discusses how to distribute a secret value d among n parties, such that at least k parties are required for restoring the secret. The idea utilizes a polynomial function $f(x) = \sum_{i=0}^{k-1} a_i x^i$,with $a_0 = d$, and distributes the values $f(i)$ to the parties. In case k of these values are commonly known, the polynomial $f(0)$ can be restored.

The advantage of this method is that the shared parts are not larger than the original data. By some deploying strategies of the parts hierarchical encryption protocols are also possible.

6.2.2 Related Approaches

The problem to protect individual privacy in a distributed scenario with an untrusted server receives increasing importance with the spread of Big Data architectures and the wide availability of massive mobility data streams. Thus, the problem is subject of many recent publications.

The work in [ABN08] computes k-anonymity and assumes a trusted server. The work from [KMM12] tries to solve the un-trusted server problem by introduction of an obfuscation layer

in the network communication, see Figure 3. But individual location data is identifying, even if it is aggregated in space-time compounds [MAA+10]. Therefore, this work still delivers the vulnerable data to the server.

Recently, differential privacy was applied to the problem in [MWP+13]. Originated in database theory, differential privacy implies that adding or deleting a single record to a database does not significantly affect the answer to a query [DMNS06b]. The work in [MWP+13] follows the common method to achieve differential privacy by adding Laplace noise (with the probability density function $Lap(\mu, \lambda) = p(x|\mu, \lambda) = \frac{1}{2\lambda} e^{-|x-\mu|/\lambda}$, where μ is set to zero and $\lambda = 1/\epsilon$) to every flow value in the vector, as proposed in [DMNS06b], compare Figure 4.

However, for cell counts differential privacy is known to provide strange behaviour, especially if large number of cells are zero [MS11]. Moreover, movement often is a routine behaviour [LKM08] and within their considered time interval most likely similar counts are produced for every person [LKM09], this offers a chance to extract the mean and thus the correct value of the distribution within a stream environment [Dua09] as the noise is sampled from $Lap(0, 1/\epsilon)$ instead of sampling from $Lap(0, m/\epsilon)$, where m denotes the expected number of queries. Additionally, movement is not random, and thus the frequencies in the vector are not independent, but correlate. Thus, combination of various noisy replies may be utilized to reveal the true distributions.

In contrast, our approach based on homomorphic cryptology in conjunction with a shared key ensures that individual data may not be accessed by the server but only aggregates of at least k people can be used, Since k may equal the number of clients, no data on the individual

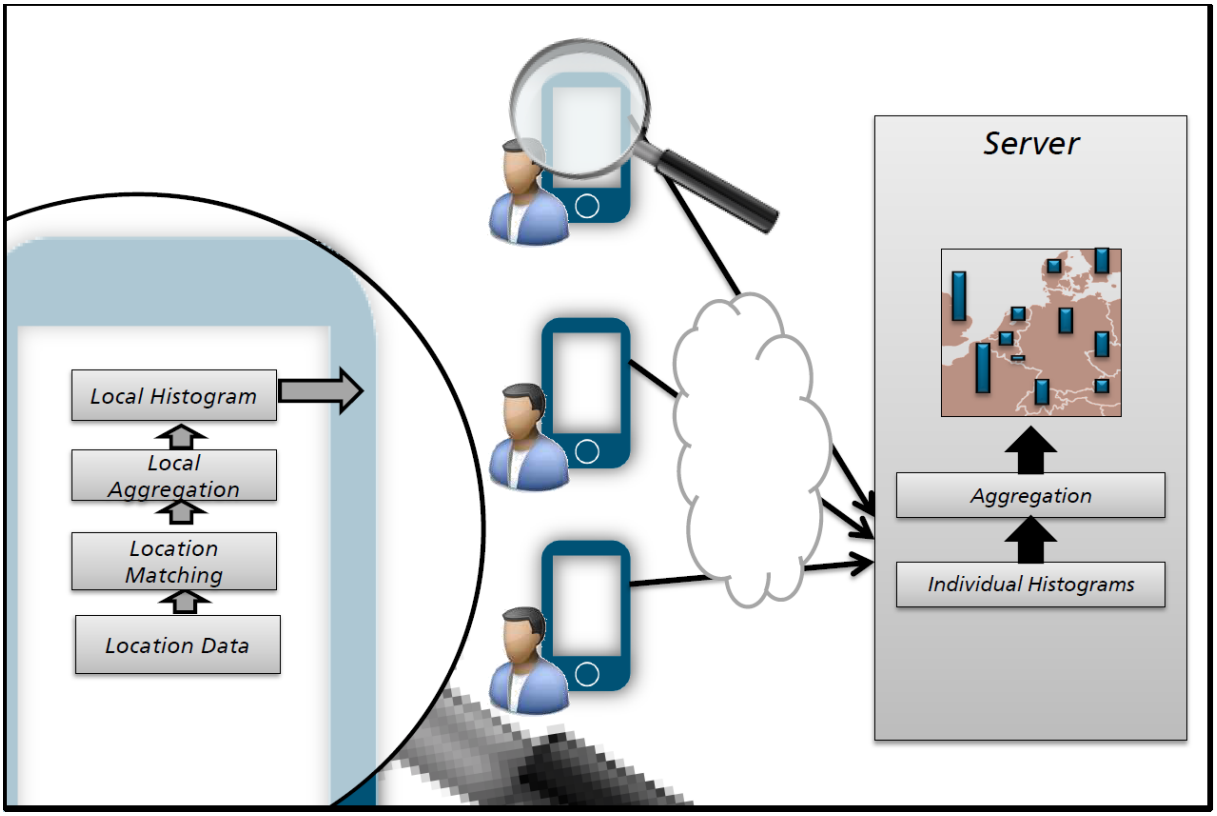


Figure 3: Obfuscated Communication in the Distributed Monitoring Scenario [KMM12].

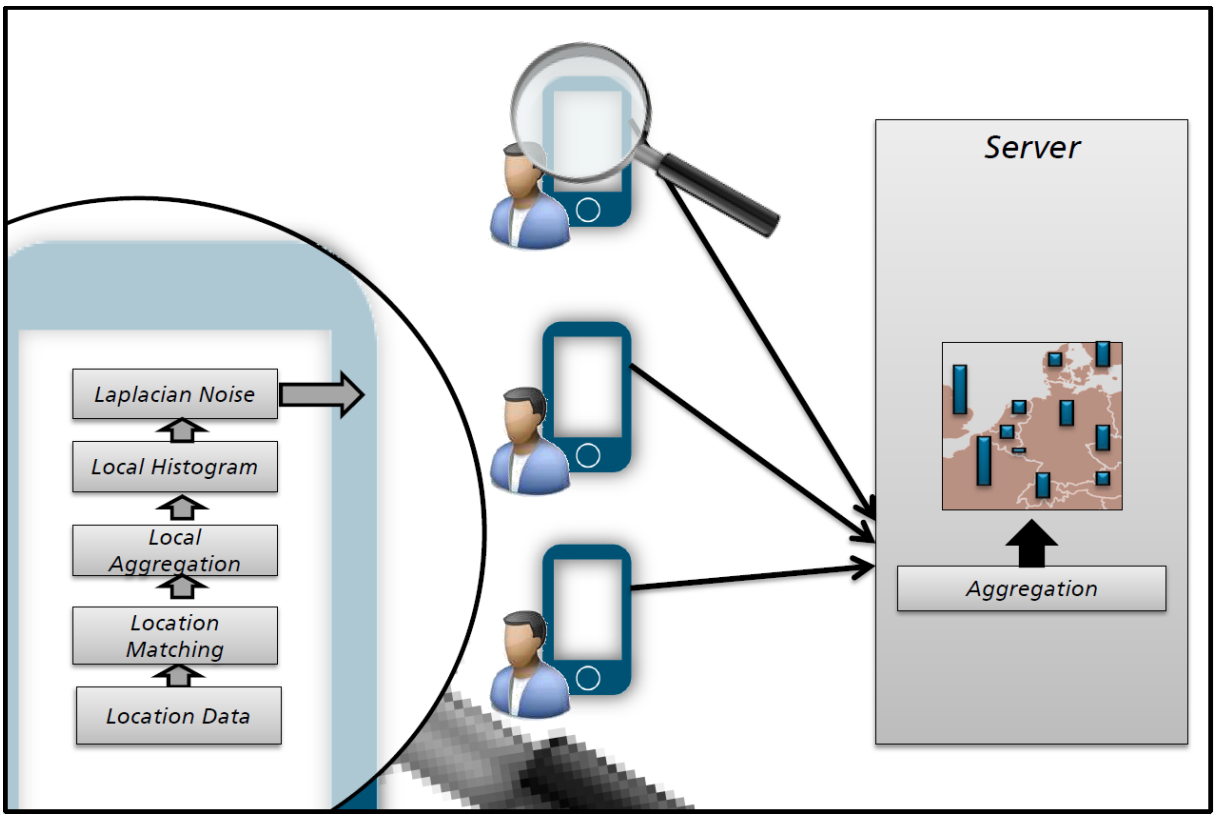


Figure 4: Differential Privacy for the Distributed Monitoring Scenario [MWP+13].

persons need to be revealed.

In contrast to previously described approaches our method (1) encrypts the values of the histogram, (2) communicates these ciphertexts to the server, (3) aggregates the ciphertexts and finally (4) decrypts the result, see an overview in Figure 5. The process utilizes asymmetric cryptography methods using two separate keys: one for encryption and another one for decryption. The utilization of a homomorphic crypto system in conjunction with Shamir's secret sharing guarantees that the individual messages can not be restored, but their sum.

6.2.3 Hash Chain

The work in [Lam81] describes a method for authentication with temporally changing password messages. The passwords series are created in advance using a cryptographic hash function which is a one-way function $F(x)$. They are created as follows $F^n(x) = F(F^{n-1}(x))$, where x is a password seed. The passwords are used in reversed order. Thus, the server stores the last value that the client sent, $F^n(x)$, and proves correctness of the new value $F^{n-1}(x)$ by verification of $F^n(x) = F(F^{n-1}(x))$. Afterwards the server stores the latest received value for the next check. As $F(\cdot)$ is a one-way function, the server may not pre-compute next password.

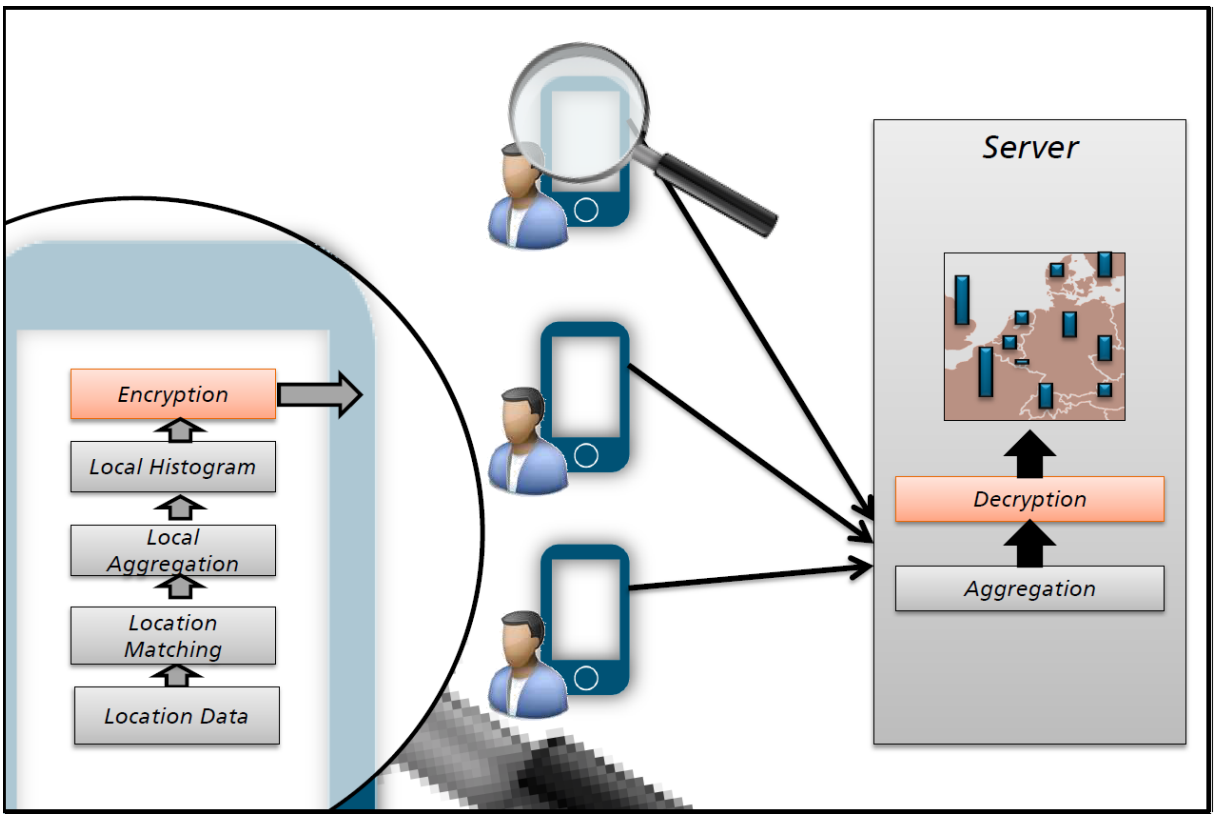


Figure 5: Proposed Privacy Preserving Aggregation of Distributed Mobility Data Streams.

6.2.4 Putting Things Together

Our cryptographic system follows the protocol of the homomorphic crypto system in [DJ01]. Consider communication among w clients with a single server. Similar to [DJ01] key generation starts with two primes p and q which are composed as $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes but different from p and q. The RSA modulus n is set to $n = pq$ and $m = p'q'$. With some decision for $s > 0$ the plaintext space becomes $\mathbb{Z}_{n^s}$. Next, d is chosen such that $d = 0 \bmod m$ and $d = 1 \bmod n^s$. Now, we use Shamir's secret sharing scheme [Sha79] to generate the private key shares of d to be divided among the clients. Thus, we apply the polynomial $f(X) = \sum_{i=0}^{w} a_i X^i \bmod l$, by picking a_i for $(0 < i \leq w)$ as random values from $0, \ldots, l$ and $a_0 = d$, l is a prime with $n^{s+1} < l$. We choose g as $g = n + 1$. The secret share of d for the i'th client will be $s_i = f(i)$. A verification key $v_i = v^{\Delta s_i} \bmod n^{s+1}$ is associated with each client i. The public key then becomes (n, s, l) and $s_1, \ldots, s_w$ is a set of private key shares.

Encryption:

The plaintext of the ith client m'_i, which is element of $\mathbb{Z}_{n^s}$, is multiplied with the one-way hash function $F^n = F(F^{n-1}(a))$ of a commonly known seed a. Thus the plaintext for the encryption results as $m_i := m'_i F^n$. Given this plaintext m_i we choose at random $r \in \mathbb{Z}^*_{n^{s+1}}$. The ciphertext $E(m_i, r)$ computes as:

$$E(m_i, r) = g^{m_i} r^{n^s} \bmod n^{s+1} \,. \tag{9}$$

The client i then communicates $c_i^{2\Delta s_i}$, with $\Delta = l!$ [DJ01].

Decryption:
The server can verify that the client raised s_i in the encryption step by testing for $log_{c_i^4}(c_i^2) = log_v(v_i)$. After the required k number of shares S arrived. They can be combined to [DJ01]:

$$c' = \prod_{i \in S} c_i^{2\lambda_{0,i}^S} \text{ mod } n^{s+1} \text{ , where} \quad (10)$$
$$\lambda_{0,i}^S = \Delta \prod_{i' \in S \setminus i} \frac{-i}{i - i'} \in \mathbb{Z} .$$

Thus, the value of c' has the form $c' = (\prod_{i \in S} c_i)^{4\Delta^2 f(0)} = (\prod_{i \in S} c_i)^{4\Delta^2 d}$. As $4\Delta^2 d = 0$ mod λ and $4\Delta^2 d = 4\Delta^2$ mod n^s, $c' = (1+n)^{4\Delta^2 \sum_{i \in S} m_i}$ mod n^{s+1}. The desired plaintext $\sum_{i \in S} m_i$ can be obtained by previously introduced algorithm and succeeding multiplication with $(4\Delta^2)^{-1}$ mod n^s. The original plaintext can be computed by dividing the resulting sum by F^n. This ensures that previous messages may not be used for analysis of current messages. The homomorphic property of the system is directly used, and bases on the work presented in [DJ01].
Security:
The security of the crypto system is based on the *decisional composite residuosity assumption* already used by [Pai99]. The assumption states that given a composite n and an integer z it is hard to decide whether z is a n-residue (i.e. a n-th power) modulo n^2, i.e. whether it exists an y with $z = y^n$ mod n^2 .

More Information

T. Liebig. Privacy preserving centralized counting of moving objects. In F. Bacao, M. Y. Santos, and M. Painho, editors, *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, pages 91–103. Springer International Publishing, 2015

T. Liebig. Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services*, pages 86–99, 2014

Both publications were published in preparation for the VaVeL project and have no VaVeL acknowledgement.

6.3 Data Mining with Order Preserving Symmetric Encryption

Public data access may stimulate research in one domain. To push research on confidential stock market indices, the company Numerai uses order preserving encryption schemes to publish their data [BCLO09, BCO11][2]. Order preserving encryption modifies the data such that the order of the original data is preserved also on the cryptotext. This allows for comparisons on the cryptotext which are a main component of many supervised data analysis tasks (regression and classification).

[2]https://medium.com/@Numerai/encrypted-data-for-efficient-markets-fffbe9743ba8#.500o56qzi, last accessed January 08th 2016

6.4 Fully Homeomorphic Encryption Scheme

Whereas Paillier's crypto scheme has an additive homeomorphic property, fully homeomorphic encryption schemes also have homeomorphic properties for multiplication. Most prominent example is Gentry's Fully Homeomorphic Encryption scheme [Gen10, VDGHV10, BGV12]. A comprehensive overview of adaptations of Gentry's Fully Homeomorphic Encryption Scheme can be found in [ABC+15]. The work in [LLAN14] uses fully homeomorphic encryption to enable private analysis on genomic data. The weaknesses of FHE in cloud based services are addressed in [CL15].

Several data mining methods are applicable on encrypted data using fully homeomorphic encryption. However, the required computation time is very high and the methods are yet of no practical relevance. In the state-of-the-art literature algorithms are described for unsupervised methods:

- **Clustering** of vertically partitioned data, [VC03],
- **Frequent Pattern Mining** The [LLXF15] bases on the fully homeomorphic encryption introduced by [VDGHV10]. Drawbacks of their method are the high complexity of the bootstrapping and other methods emerged [AEH15]. A novel FHE method without bootstrapping is presented in [BGV12].

and supervised data mining methods:

- **Regression** [WH12],
- **Classification** [BPTG14],
- **Outlier detection** of vertically partitioned data [VC03].

Besides Data Mining, also search and optimization is an import topic in Artificial Intelligence and especially trip computations in a traffic network are of high interest to the VAVEL project. However, the encrypted routing method in [Gue13] uses a proactive distributed (table) routing and incorporates Gentry's fully homeomorphic encryption. Their routing scheme, however does not disclose the routing request from the routing engine, but is a distributed routing approach in communication networks and this setting differs, for example, in the point that messages may get doubled to reach a certain goal.

7 Privacy via Secret Sharing

Previous section already introduced the multi-party computing scenario. We presented how the secret key of a asymmetric encryption scheme may be distributed amongst multiple parties. In a scenario where the data should be clustered, it is possible to ensure data privacy just by sharing a secret. Next, we present two exemplified algorithms for privacy preserving clustering using secret sharing. The first one focuses on vertically partitioned data, whereas the second focuses on horizontally partitioned data.

7.1 Distributed Clustering on Vertically Partitioned Sata based on Secret Sharing

A possible method to cluster vertically distributed data with secure multi party computation was discussed in Section 6.4. There, we presented the method [VC03] which utilizes homeomorphic encryption for clustering distributed data. These computations are computationally expensive. In contrast, we describe here [DPS+08] that uses secret sharing. Secret sharing partitions the data over multiple parties and ensures in this way that no single party may reconstruct any information. The algorithm in [DPS+08] involves r parties. Since data is vertically distributed, the n data entities are distributed such that every party holds just a fraction of the attributes. Aim of the proposed method to perform k-means clustering in this setting without revealing the data to the other parties. The method assigns the entities to the clusters, and the parties retrieve the indices of the clusters with at least one entity. Additionally, the parties get a list of cluster-means that correspond to the own attributes.

Initially, their algorithm assigns a random cluster to every entity, and every entity is assigned to the closed cluster. Afterwards the cluster means are updated. Last two steps are iterated until there is no change in cluster. Challenging is the computation of the distances to the cluster mean, as the partial Euclidean distances (that a party may easily compute) may not be communicated since it contains vulnerable information. Solution is the pairwise secret sharing of the vulnerable distances: The local component of some distance among an entity to a cluster mean is split into a sum over all other parties, and every party receives a part of this information as a single summand. This is done for all parties. In the end the aggregated distance matrix can be revealed using all of the secret shares without revealing which impact which party had.

7.2 Distributed Clustering on Horizontally Partitioned Data based on Secret Sharing

Secret Sharing describes for t involved parties, that the data can just be accessed if t parties are involved. For any smaller number $x < t$ of parties it is not possible to make decisions based on the data.

The proposed protocol in [KPSS07] distributes the data amongst two parties that prepare distance computation and then centralizes it to compute the distance matrix. Thus, it decouples the data holder from the dataminer that may not access the original data, by introduction of a trusted multi party network that prepares the data accordingly. The method bases on sharing a secret number s in two parts, a random one r and $s - r$ amongst the parties. Addition of the two values reveals the original data, but for that operation both parties are required.

To prepare computation of the distance matrix for clustering (in this case just Manhattan distance is supported), the collected shares at a particular party are pairwise subtracted and some pseudo random method assigns the sign (to obfuscate origin of the data afterwards). The data becomes aggregated and is distances can be computed without reveling the data to the parties.

8 Summary

In this deliverable we offered a brief introduction to privacy preserving data-mining methods for mobility data and highlighted the contributions of the VaVeL consortium. It is worth mentioning that especially the cryptographic approaches tend to slow down the analysis process. In addition, also the energy consumption of a privacy-preserving data analysis system increases [PRRJ06]. To overcome these challenges, the authors in [GDC98] presented an energy scalable encryption processor that preserves privacy on processor level.

Most work on privacy attempts to limit disclosure risk: the probability that some adversary can link a released record to a particular member of the population or identify that someone belongs to a dataset that generates a statistic [DL86, Rei05, KKO+06]. In statistical literature, work on disclosure limitation and so-called linkage risk, for example as in the framework of Duncan and Lambert [DL86], has yielded several techniques for maintaining privacy, such as aggregation, swapping features or responses between different data points, or perturbation of data. Other authors have proposed measures for measuring the utility of released data (e.g., [KKO+06, CKK11]). The most prominent measure of privacy is differential privacy, due to [DMNS06a], which roughly states that the answer to a data query must not depend too much on the samples, and it should be difficult, given the answer to a query, to ascertain whether a vector is contained in the used dataset.

Furthermore, we presented the contributions of the VaVeL consortium to privacy-by-design analyses of mobility data. Thus we highlighted the increased re-identification risks in mobility data. We highlighted methods aggregation of individual trajectories and for decentralized prediction based on aggregated data and a privacy preserving method for crowdsourcing.

The methods presented so far may be categorized (using the scheme proposed by [VS16], compare Section 2.4) as shown in following list:

- **Minimize** individual data should be restricted to least possible quantity.

  This is achieved by aggregation methods (Section 3) and data perturbation methods (Section 4).

- **Hide** Private data must be concealed from unauthorized view.

  This is achieved by perturbation methods (Section 4), sketches (Section 5), and secure multiparty computing: homeomorphic encrypted methods (Section 6) and secret sharing (Section 7).

- **Separate** Private data must be interpreted in separate partitions.

  This is achieved in secure multiparty computing methods (Sections 6 and 7).

- **Aggregate** Private data should be treated with a better level of aggregation.

  This is achieved in secure multiparty computing methods (Sections 6 and 7) as well as aggregation and perturbation methods (Sections 3 and 4).

However, data privacy can not be guaranteed without knowing the purpose of the analysis, this is why lawyers and computer scientists need to collaborate on this subject [WS14]. This deliverable focussed on the algorithmic challenges and our contributions to privacy-preserving data analysis, however, methods for the last properties from Section 2.4 (**Inform**, **Control**, **Enforce**, and **Demonstrate**) can be found in Section 4.3 in [VS16].

The methods we presented have impact on the analysis methods in D5.1 and D4.2, and we will further investigate how to protect individual privacy in our analyses. This will be reflected in D5.2 and D5.3.

References

[AAS+12] N. Andrienko, G. Andrienko, H. Stange, T. Liebig, and D. Hecker. Visual analytics for understanding spatial situations from episodic movement data. *KI - Künstliche Intelligenz*, pages 241–251, 2012.

[ABC+15] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter, and M. Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192, 2015. `http://eprint.iacr.org/`.

[ABN08] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 376–385, Washington, DC, USA, 2008. IEEE Computer Society.

[AEH15] L. J. Aslett, P. M. Esperança, and C. C. Holmes. A review of homomorphic encryption and software tools for encrypted statistical machine learning. *arXiv preprint arXiv:1508.06574*, 2015.

[AGDG+13] G. Andrienko, A. Gkoulalas-Divanis, M. Gruteser, C. Kopp, T. Liebig, and K. Rechert. Report from dagstuhl: the liberation of mobile location data and its implications for privacy research. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(2):7–18, 2013.

[BCLO09] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *Advances in Cryptology-EUROCRYPT 2009*, pages 224–241. Springer, 2009.

[BCO11] A. Boldyreva, N. Chenette, and A. ONeill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *Advances in Cryptology–CRYPTO 2011*, pages 578–595. Springer, 2011.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[BHV10] M.-F. Balcan, S. Hanneke, and J. W. Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2–3):111–139, 2010.

[Bil11a] N. Bilton. 3G Apple iOS devices are storing users location data. The New York Times, Published: April 20, 2011, 2011.

[Bil11b] N. Bilton. Holding companies accountable for privacy breaches. The New York Times, Published: April 27, 2011, 2011.

[BK13] I. Boutsis and V. Kalogeraki. Privacy preservation for participatory sensing data. *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 0:103–113, 2013.

[BK16] I. Boutsis and V. Kalogeraki. Location privacy for crowdsourcing applications. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 694–705, New York, NY, USA, 2016. ACM.

[BLR13] A. Blum, K. Ligett, and A. Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[BPTG14] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. *Crypto ePrint Archive*, 2014.

[CKD+04] C. Clifton, M. Kantarcioglu, A. Doan, G. Schadow, J. Vaidya, A. K. Elmagarmid, and D. Suciu. Privacy-preserving data integration and sharing. In *DMKD*, pages 19–26, 2004.

[CKK11] L. H. Cox, A. F. Karr, and S. K. Kinney. Risk-utility paradigms for statistical disclosure limitation: How to think, but not how to act. *International Statistical Review*, 79(2):160–183, 2011.

[CL15] Z. Cao and L. Liu. On the weakness of fully homomorphic encryption. *CoRR*, abs/1511.05341, 2015.

[CLQZ09] S. Chen, B. Liu, M. Qian, and C. Zhang. Kernel k-Means based framework for aggregate outputs classification. In *Proc. of the Int. Conf. on Data Mining Workshops (ICDMW)*, pages 356–361, 2009.

[CM05] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[CMS11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *The Journal of Machine Learning Research*, 12:1069–1109, 2011.

[CMY08] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1076–1085. Society for Industrial and Applied Mathematics, 2008.

[CSZ06] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[CY16] Y. Cao and M. Yoshikawa. Differentially private real-time data publishing over infinite trajectory streams. *IEICE TRANSACTIONS on Information and Systems*, 99(1):163–175, 2016.

[DBV11] K. Das, K. Bhaduri, and P. Votava. Distributed anomaly detection using 1-class SVM for vertically partitioned data. *Stat. Anal. Data Min.*, 4(4):393–406, 2011.

[DDFS16] M. Douriez, H. Doraiswamy, J. Freire, and C. T. Silva. Anonymizing nyc taxi data: Does it matter? 2016.

[DJ01] 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, pages 119–136, London, UK, UK, 2001. Springer-Verlag.

[DL86] G. T. Duncan and D. Lambert. Disclosure-limited data dissemination. *Journal of the American statistical association*, 81(393):10–18, 1986.

[DMNS06a] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[DMNS06b] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[DN03] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210. ACM, 2003.

[DPS+08] M. C. Doganay, T. B. Pedersen, Y. Saygin, E. Savaş, and A. Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, pages 3–11. ACM, 2008.

[Dua09] Y. Duan. Privacy without noise. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1517–1520, New York, NY, USA, 2009. ACM.

[Dwo08] C. Dwork. Differential privacy: A survey of results. In *Theory and applications of models of computation*, pages 1–19. Springer, 2008.

[FFGM07] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Analysis of Algorithms 2007 (AofA07)*, pages 127–146, 2007.

[FMKM12] S.-C. Florescu, M. Mock, C. Körner, and M. May. Efficient Mobility Pattern Detection on Mobile Devices. In *Proceedings of the ECAI'12 Workshop on Ubiquitous Data Mining*, pages 23–27, 2012.

[FZY+14] K. Fan, H. Zhang, S. Yan, L. Wang, W. Zhang, and J. Feng. Learning a generative classifier from label proportions. *Neurocomput.*, 139:47–55, 9 2014.

[GDC98] J. Goodman, A. P. Dancy, and A. P. Chandrakasan. An energy/security scalable encryption processor using an embedded variable voltage dc/dc converter. *IEEE Journal of Solid-State Circuits*, 33(11):1799–1809, 1998.

[Gen10] C. Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.

[GFTB17] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs. $k^{\tau,\epsilon}$-anonymity: Towards privacy-preserving publishing of spatiotemporal trajectory data. *arXiv preprint arXiv:1701.02243*, 2017.

[GKS08] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2008.

[GLN13] T. Graepel, K. Lauter, and M. Naehrig. Ml confidential: Machine learning on encrypted data. In *Information Security and Cryptology–ICISC 2012*, pages 1–21. Springer, 2013.

[GP08] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.

[GP09] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In H. Tokuda, M. Beigl, A. Friday, A. Brush, and Y. Tobe, editors, *Pervasive Computing*, volume 5538 of *Lecture Notes in Computer Science*, pages 390–397. Springer Berlin / Heidelberg, 2009.

[Gue13] A. Guellier. Homomorphic cryptography based anonymous routing. cryptography and security [cs.cr] ¡dumas-00854815¿. 2013. [Online; accessed 15-Dezember-2015].

[Hel11] M. Helft. Apple and Google use phone data to map the world. The New York Times, Published: April 25, 2011, 2011.

[HGIL13] J. Hernndez-Gonzlez, I. Inza, and J. A. Lozano. Learning bayesian network classifiers from label proportions. *Pattern Recognition*, 46(12):3425–3440, 2013.

[HHJ+11] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 639–652, New York, NY, USA, 2011. ACM.

[HIJ+12] B. Hoh, T. Iwuchukwu, Q. Jacobson, D. B. Work, A. M. Bayen, R. Herring, J. C. Herrera, M. Gruteser, M. Annavaram, and J. Ban. Enhancing Privacy and Accuracy in Probe Vehicle-Based Traffic Monitoring via Virtual Trip Lines. *IEEE Trans. Mob. Comput.*, 11(5):849–864, 2012.

[HON+12] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang. Accomplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–9. IEEE, 2012.

[HPM14] S. Hess, N. Piatkowski, and K. Morik. Shrimp: Descriptive patterns in a tree. In *LWA*, pages 181–192, 2014.

[HQTK16] A. Hossain, A. Quattrone, E. Tanin, and L. Kulik. On the effectiveness of removing location information from trajectory data for preserving location privacy. In *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 49–54. ACM, 2016.

[HRW11] R. Hall, A. Rinaldo, and L. Wasserman. Random differential privacy. *arXiv preprint arXiv:1112.2680*, 2011.

[Joh84] D. S. Johnson. The NP-completeness column: An ongoing guide . *Journal of Algorithms*, 5(3):433–447, 1984.

[KdF05] H. Kück and N. de Freitas. Learning to classify individuals based on group statistics. In *Proc. of the 21th UAI*, pages 332–339, 2005.

[KKM+13] M. Kamp, C. Kopp, M. Mock, M. Boley, and M. May. Privacy-preserving mobility monitoring using sketches of stationary sensor readings. In *Machine Learning and Knowledge Discovery in Databases*, pages 370–386. Springer, 2013.

[KKO+06] A. F. Karr, C. N. Kohnen, A. Oganian, J. P. Reiter, and A. P. Sanil. A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, 60(3):224–232, 2006.

[KMM12] C. Kopp, M. Mock, and M. May. Privacy-preserving distributed monitoring of visit quantities. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 438–441, New York, NY, USA, 2012. ACM.

[KNW10] D. M. Kane, J. Nelson, and D. P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 41–52. ACM, 2010.

[KPSS07] S. V. Kaya, T. B. Pedersen, E. Savaş, and Y. Saygıỳn. Efficient privacy preserving distributed clustering based on secret sharing. In *Emerging Technologies in Knowledge Discovery and Data Mining*, pages 280–291. Springer, 2007.

[Lam81] L. Lamport. Password authentication with insecure communication. *Commun. ACM*, 24(11):770–772, November 1981.

[Lie14] T. Liebig. Privacy preserving aggregation of distributed mobility data streams. In *Proceedings of the 11th Symposium on Location-Based Services*, pages 86–99, 2014.

[Lie15a] T. Liebig. Das Paillier Cryptosystem mit Beispielen in Cran-R. http://www-ai.cs.uni-dortmund.de/LEHRE/FACHPROJEKT/WS1516/download/paillier.pdf, 2015. [Online; accessed 14-Dezember-2016].

[Lie15b] T. Liebig. Privacy preserving centralized counting of moving objects. In F. Bacao, M. Y. Santos, and M. Painho, editors, *AGILE 2015*, Lecture Notes in Geoinformation and Cartography, pages 91–103. Springer International Publishing, 2015.

[Lie15c] T. Liebig. Privacy preserving centralized counting of moving objects. In *AGILE 2015*, pages 91–103. Springer, 2015.

[Lie16] T. Liebig. Ai-based analysis methods in spatio-temporal data mining. In M. Jankowska, M. Pawelczyk, S. Allouche, and M. Kulawiak, editors, *AI: Philosophy, Geoinformatics & Law*, pages 133–150. IUS PUBLICUM, Warsaw, 2016.

[Lie17] T. Liebig. Smart navigation - chances, risk and challenges. In M. Jankowska, M. Pawelczyk, S. Augustyn, and M. Kulawiak, editors, *Navigation and Earth Observation - Law & Technology*, page (in press). IUS PUBLICUM, Warsaw, 2017.

[LKM08] T. Liebig, C. Körner, and M. May. Scalable sparse bayesian network learning for spatial applications. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 420–425. IEEE, 2008.

[LKM09] T. Liebig, C. Körner, and M. May. Fast visual trajectory analysis using spatial bayesian networks. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 668–673. IEEE, 2009.

[LKR06] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106, 2006.

[LKW12] T. Liebig and A. U. Kemloh Wagoum. Modelling microscopic pedestrian mobility using bluetooth. In *ICAART*, pages 270–275. SciTePress, 2012.

[LLAN14] K. Lauter, A. López-Alt, and M. Naehrig. Private computation on encrypted genomic data. In *Progress in Cryptology-LATINCRYPT 2014*, pages 3–27. Springer, 2014.

[LLXF15] J. Liu, J. Li, S. Xu, and B. C. Fung. Secure outsourced frequent pattern mining by fully homomorphic encryption. In *Big Data Analytics and Knowledge Discovery*, pages 70–81. Springer, 2015.

[LSM12] S. Lee, M. Stolpe, and K. Morik. Separable approximate optimization of support vector machines for distributed sensing. In *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *LNCS*, pages 387–402, Berlin, Heidelberg, 2012. Springer-Verlag.

[LSM15] T. Liebig, M. Stolpe, and K. Morik. Distributed traffic flow prediction with label proportions: From in-network towards high performance computation with mpi. In G. Andrienko, D. Gunopulos, I. Katakis, T. Liebig, S. Mannor, K. Morik, and F. Schnitzler, editors, *Proceedings of the 2nd International Workshop on Mining*

Urban Data (MUD2), volume 1392 of *CEUR Workshop Proceedings*, pages 36–43. CEUR-WS, 2015.

[LXM13] T. Liebig, Z. Xu, and M. May. Incorporating Mobility Patterns in Pedestrian Quantity Estimation and Sensor Placement. In J. Nin and D. Villatoro, editors, *Proceedings of the First International Workshop on Citizen Sensor Networks CitiSens 2012, LNAI 7685*, pages 67–80. Springer, 2013.

[MAA+10] A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Journal of Transactions on Data Privacy*, 3(2):91–121, 2010.

[McC11] D. McCullagh. Microsoft collects locations of Windows phone users. CNet News, Published: April 25, 2011, 2011.

[McC14] B. McCann. A review of scats operation and deployment in dublin. In *Proceedings of the 19th JCT Traffic Signal Symposium & Exhibition*. JCT Consulting Ltd, 2014.

[MCO07] D. R. Musicant, J. M. Christensen, and J. F. Olson. Supervised learning by training on aggregate outputs. In *7th Int. Conf. on Data Mining (ICDM)*, pages 252–261, 10 2007.

[MDDC15] L. Melis, G. Danezis, and E. De Cristofaro. Efficient private statistics with succinct sketches. *arXiv preprint arXiv:1508.06110*, 2015.

[MPPP14] A. Monreale, D. Pedreschi, R. G. Pensa, and F. Pinelli. Anonymity preserving sequential pattern mining. *Artificial intelligence and law*, 22(2):141–173, 2014.

[MS11] K. Muralidhar and R. Sarathy. Does differential privacy protect terry gross privacy? In J. Domingo-Ferrer and E. Magkos, editors, *Privacy in Statistical Databases*, volume 6344 of *Lecture Notes in Computer Science*, pages 200–209. Springer Berlin Heidelberg, 2011.

[MT07] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[MVBC12] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 323–336. ACM, 2012.

[MWP+13] A. Monreale, W. Wang, F. Pratesi, S. Rinzivillo, D. Pedreschi, G. Andrienko, and N. Andrienko. Privacy-preserving distributed movement data aggregation. In *Geographic Information Science at the Heart of Europe*, Lecture Notes in Geoinformation and Cartography, pages 225–245. Springer International Publishing, 2013.

[MYYR10] C. Y. T. Ma, D. K. Y. Yau, N. K. Yip, and N. S. V. Rao. Privacy vulnerability of published anonymous mobility traces. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 185–196, New York, NY, USA, 2010. ACM.

[Ohm09] P. Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review, Vol. 57, p. 1701, 2010*, 2009.

[O'K08] M. O'Keeffe. The paillier cryptosystem. *A Look Into The Cryptosystem And Its Potential Application, college of New Jersey*, 2008.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology?EUROCRYPT?99*, pages 223–238. Springer, 1999.

[PDCR16] A. Pyrgelis, E. De Cristofaro, and G. Ross. Privacy-friendly mobility analytics using aggregate location data. *arXiv preprint arXiv:1609.06582*, 2016.

[PK01] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymitya proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[PNCR14] G. Patrini, R. Nock, T. Caetano, and P. Rivera. (almost) no label no cry. In *Advances in Neural Information Processing Systems 27*, pages 190–198. Curran Associates, Inc., 2014.

[PRRJ06] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on mobile computing*, 5(2):128–143, 2006.

[PTDC17] A. Pyrgelis, C. Troncoso, and E. De Cristofaro. What does the crowd say about you? evaluating aggregation-based location privacy. *arXiv preprint arXiv:1703.00366*, 2017.

[QSCL09] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *J. Mach. Learn. Res.*, 10:2349–2374, 12 2009.

[RBHT09] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *arXiv preprint arXiv:0911.5708*, 2009.

[Rei05] J. P. Reiter. Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association*, 100(472):1103–1112, 2005.

[RSA83] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26(1):96–99, January 1983.

[Rüp10] S. Rüping. SVM classifier estimation from group probabilities. In *Proc. of the 27th Int. Conf. on Machine Learning (ICML)*, pages 911–918, 2010.

[SBDM13] M. Stolpe, K. Bhaduri, K. Das, and K. Morik. Anomaly detection in vertically partitioned data by distributed core vector machines. In *European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 321–336. Springer, 2013.

[SCR+11] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(22):612–613, 1979.

[Sie16] A. Siebes. Sharing data with guaranteed privacy. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 96–108. Springer, 2016.

[SL16] P. Sui and X. Li. Roat: Road-network-based anonymization of trajectories. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*, pages 309–314. IEEE, 2016.

[SLM15] M. Stolpe, T. Liebig, and K. Morik. Communication-efficient learning of traffic flow in a network of wireless presence sensors. In *Proceedings of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases (PDCKDD 2015)*, CEUR Workshop Proceedings. CEUR-WS, 2015.

[SM11] M. Stolpe and K. Morik. Learning from label proportions by optimizing cluster model selection. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, ECML PKDD'11, pages 349–364, Berlin, Heidelberg, 2011. Springer-Verlag.

[Ste10] A. Steffen. The paillier cryptosystem. http://slideplayer.com/slide/8488065/, 2010. [Online; accessed 14-Dezember-2015].

[Swe02] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[TPMS16] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos. Local suppression and splitting techniques for privacy preserving publication of trajectories. *Transactions on Knowledge and Data Engineering*, 2016.

[VC03] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2003.

[VDGHV10] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.

[VS16] J. Vinothkumar and V. Santhi. A study on privacy preserving methodologies in big data. *Indian Journal of Science and Technology*, 9(S1), 2016.

[VVLS11] J. Vreeken, M. Van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, 23(1):169–214, 2011.

[War65] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[WBCL16] F.-J. Wu, M. R. Brust, Y.-A. Chen, and T. Luo. The privacy exposure problem in mobile location-based services. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–7. IEEE, 2016.

[WH12] D. Wu and J. Haven. Using homomorphic encryption for large scale statistical analysis. 2012.

[WJD12] M. J. Wainwright, M. I. Jordan, and J. C. Duchi. Privacy aware learning. In *Advances in Neural Information Processing Systems*, pages 1430–1438, 2012.

[WS14] P. Weiser and S. Scheider. A civilized cyberspace for geoprivacy. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Privacy in Geographic Information Collection and Analysis*, GeoPrivacy '14, pages 5:1–5:8, New York, NY, USA, 2014. ACM.

[WSN16] S. Wang, R. Sinnott, and S. Nepal. Privacy-protected social media user trajectories calibration. In *e-Science (e-Science), 2016 IEEE 12th International Conference on*, pages 293–302. IEEE, 2016.

[WZ10] L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[YKJC14] F. X. Yu, S. Kumar, T. Jebara, and S. Chang. On learning with label proportions. *CoRR*, abs/1402.5902, 2014.

[YLK+13] F. X. Y. Yu, D. Liu, S. Kumar, T. Jebara, and S. Chang. $\propto$SVM for learning with label proportions. In *Proc. of the 30th Int. Conf. on Machine Learning (ICML)*, pages 504–512, 2013.

[ZB11] 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*, MobiCom '11, pages 145–156, New York, NY, USA, 2011. ACM.

[ZWL08] S. Zhou, L. Wasserman, and J. D. Lafferty. Compressed regression. In *Advances in Neural Information Processing Systems*, pages 1713–1720, 2008.