Quality & Privacy in User-generated Big Data: Algorithms & Techniques

Manos Katsomallos

Quality & Privacy in User-generated Big Data: Algorithms & Techniques

PRÉSENTÉE LE December 13, 2021

À LA CY TECH - SCIENCES ET TECHNIQUES
EQUIPES TRAITEMENT DE L'INFORMATION ET SYSTÈMES (ETIS)
PROGRAMME DOCTORAL EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION (STIC)

CY CERGY PARIS UNIVERSITÉ

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Manos KATSOMALLOS

acceptée sur proposition du jury :

Dimitris KOTZINOS, PR, CY Cergy Paris Université, Directeur de thèse
Katerina TZOMPANAKI, MCF, CY Cergy Paris Université, Co-encadrante de thèse
Benjamin NGUYEN, PR, INSA Centre Val de Loire, Rapporteur
Kostas CHATZIKOKOLAKIS, MCF, Université d'Athènes, Rapporteur
Salima BENBERNOU, PR, Université de Paris, Examinatrice
Dan VODISLAV, PR, CY Cergy Paris Université, Examinateur

France
2021

Abstract

Sensors, portable devices, and crowdsensing applications generate massive amounts of user-related, and usually geo-tagged, data on a daily basis. The manipulation of such data is useful in numerous application domains including traffic monitoring, intelligent building, and healthcare. A high percentage of these data carry information of user activities and other personal details, and thus their manipulation and sharing raise concerns about the privacy of the individuals involved. To enable the secure—from the user privacy perspective—data sharing, researchers have already proposed various seminal techniques for the protection of user privacy while accounting for data utility and quality. However, the continuous fashion in which data are generated nowadays and the high availability of external sources of information, pose more threats and add extra challenges to the problem due to the inevitable presence of data correlation. It is therefore essential to design solutions that guarantee sufficient user privacy protection and maximize data utility, while providing configurability by considering the context and user preferences.

Initially, we study the literature regarding data privacy in continuous data publishing, and report on the proposed solutions, with a special focus on solutions concerning location or geo-referenced data. As a matter of fact, a wealth of algorithms has been proposed for privacy-preserving data publishing, either for microdata or statistical data. In this context, we seek to offer a guide that would allow readers to choose the proper algorithm(s) for their specific use case accordingly. We provide an insight into time-related properties of the algorithms, e.g., if they work on finite or infinite data, or if they take into consideration any underlying type of data correlation.

Thereafter, we proceed to propose a novel type of data privacy, called *landmark privacy*. We observe that in continuous data publishing, events are not equally significant in terms of privacy, and hence they should affect the privacy-preserving processing differently. Differential privacy is a well-established paradigm in privacy-preserving time series publishing. The existing differential privacy protection levels protect either a single timestamp, or all the data per user or per window in the time series; however, considering all timestamps as equally significant. The novel notion that we propose, landmark privacy, is based on

differential privacy and allocates the available privacy budget at each timestamp while taking into account significant events (*landmarks*) in the time series. This allows for better data utility by optimizing the privacy budget allocation, and thus avoiding the injection of unnecessary noise into the data releases. We design three landmark privacy schemes and further extend them by enhancing the privacy protection of the landmark set with the design of a dummy landmark selection module that renders the actual landmarks indistinguishable with the addition of regular events to the landmark set.

Finally, we evaluate the proposed landmark privacy schemes and dummy landmark selection module on real and synthetic data sets. We assess the impact on data utility for several possible landmark distributions, with emphasis on situations under the presence of temporal correlation. Overall, the results of the experimental evaluation and comparative analysis of landmark privacy validate its applicability to several use case scenarios and showcase the improvement, in terms of data utility, over the existing privacy protection levels. Particularly, the dummy landmark selection module achieves better landmark protection, provoking only a minor data utility decline. In terms of temporal correlation, we observe that under moderate and strong correlation, greater average regular–landmark event distance causes greater overall privacy loss.

Keywords: data quality, data privacy, continuous data publishing, crowdsensing, privacy-preserving data processing

Résumé

Les capteurs, les appareils portables et les applications crowdsensing génèrent quotidiennement des quantités massives de données, généralement géolocalisées, liées aux utilisateurs. La manipulation de ces données est utile dans de nombreux domaines d'application, notamment la surveillance du trafic, les bâtiments intelligents, et la santé. Un pourcentage élevé de ces données contiennent des informations sur les activités des utilisateurs et d'autres détails personnels, et donc leur manipulation et leur partage soulèvent des inquiétudes quant à la confidentialité des personnes concernées. Pour permettre le partage sécurisé—du point de vue de la confidentialité des utilisateurs—des données, les chercheurs ont déjà proposé diverses techniques fondamentales pour la protection de la confidentialité des utilisateurs tout en tenant compte de l'utilité et de la qualité des données. Cependant, la manière continue avec laquelle les données sont générées et la haute disponibilité des sources d'information externes, posent plus de menaces et ajoutent des défis supplémentaires au problème en raison de la présence inévitable de la corrélation des données. Il est donc essentiel de concevoir des solutions qui garantissent une protection suffisante de la confidentialité des utilisateurs et maximisent l'utilité des données, tout en offrant une configurabilité en tenant compte du contexte et des préférences des utilisateurs.

Initialement, nous étudions la littérature concernant la confidentialité des données dans la publication de données en continu, et rapportons les solutions proposées, avec un accent particulier sur les solutions concernant la localisation ou les données géo-référencées. En fait, une multitude d'algorithmes ont été proposés pour la publication de données préservant la confidentialité, que ce soit pour des microdonnées ou des données statistiques. Dans ce contexte, nous cherchons à offrir un guide qui permettrait aux lecteurs de choisir en conséquence le ou les algorithmes appropriés pour leur cas d'utilisation spécifique. Nous donnons un aperçu des propriétés temporelles des algorithmes, par exemple, e.g., s'ils fonctionnent sur des données finies ou infinies, ou s'ils prennent en considération tout type sous-jacent de corrélation de données.

Par la suite, nous proposons un nouveau type de confidentialité des données, appelé *confidentialité landmark*. Nous observons que dans la publication de don-

nées en continu, les événements ne sont pas aussi importants les uns aux autres en termes de confidentialité et devraient donc affecter différemment le traitement préservant la confidentialité. La confidentialité différentielle est un paradigme bien établi dans la publication de séries temporelles préservante la confidentialité. Les niveaux de protection existants de la confidentialité différentielle protègent soit un seul horodatage, soit toutes les données par utilisateur ou par fenêtre dans la série temporelle ; cependant, en considérant tous les horodatages comme également significatifs. La nouvelle notion que nous proposons, confidentialité landmark, est basée sur la confidentialité différentielle et alloue le budget de confidentialité disponible à chaque horodatage tout en tenant compte des événements significatifs (*landmarks*) dans la série temporelle. Cela permet une meilleure utilité des données en optimisant l'allocation du budget de confidentialité et en évitant ainsi l'injection de bruit inutile dans les publications de données. Nous concevons trois schémas de confidentialité landmark et les étendons davantage en améliorant la protection de la confidentialité de l'ensemble landmark avec la conception d'un module de sélection de landmark factice (dummy) qui rend les landmarks réels indiscernables avec l'ajout d'événements réguliers à l'ensemble de landmarks.

Enfin, nous évaluons les schémas de confidentialité landmark proposés et le module de sélection de landmarks factices sur des ensembles de données réelles et synthétiques. Nous évaluons l'impact sur l'utilité des données pour plusieurs distributions de landmarks possibles, en mettant l'accent sur les situations en présence de corrélation temporelle. Dans l'ensemble, les résultats de l'évaluation expérimentale et de l'analyse comparative de la confidentialité landmark valident son applicabilité à plusieurs scénarios de cas d'utilisation et montrent l'amélioration, en termes d'utilité des données, par rapport aux niveaux de protection de la confidentialité existants. En particulier, le module de sélection de landmark factice assure une meilleure protection landmark, provoquant seulement une baisse mineure de l'utilité des données. En termes de corrélation temporelle, nous observons que sous une corrélation modérée et forte, une distance moyenne plus grande entre les événements réguliers et landmark entraîne une perte globale de confidentialité plus importante.

Mots clés : confidentialité des données, qualité des données, publication continue des données, crowdsensing, traitement des données préservant la confidentialité, correlation temporelle

Acknowledgements

Upon the completion of my thesis, I would like to express my deep gratitude to Professor Dimitris Kotzinos for believing in me and for providing me with opportunities that helped me pave my path in academia.

This thesis would not have been possible without the patient guidance of Associate Professor Katerina Tzompanaki. Her love for knowledge and hard work were inspiring and served as the catalyst for every single step that I made towards getting a better grasp on computer science.

I am genuinely grateful to the reviewers of my thesis, Professor Benjamin Nguyen and Associate Professor Kostas Chatzikokolakis for their time, effort, and valuable feedback. Moreover, I would like to thank Professor Salima Benbernou and Professor Dan Vodislav for being part of my examining committee.

A special thanks goes to Alexandros Kontarinis for being an exemplary colleague and a unique companion during this journey. I would also like to thank the faculty of the Computer Sciences department of the CY Cergy Paris University, the people of the ETIS laboratory, and especially the members of the MIDI group for creating a pleasant and creative environment.

Last but not least, I wish to express my thankfulness to my family and friends for their unconditional support and encouragement all these years.

Pontoise, December 13, 2021

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

Data privacy is becoming an increasingly important issue, both at a technical and at a societal level, and introduces various challenges ranging from the way we share and publish data sets to the way we use online and mobile services. Personal information, also described as *microdata*, acquired increasing value and is in many cases used as the 'currency' [eco16] to pay for access to various services, i.e., users are asked to exchange their personal information with the service provided. This is particularly true for many *Location-Based Services* (LBSs), e.g., Google Maps [gma21], Waze [waz21], etc. These services exchange their 'free' service with collecting and using user-generated data, such as timestamped geolocalized information. Besides navigation and location-based services, social media applications, e.g., Facebook [fac21], Twitter [twi21], Foursquare [fou21], etc. take advantage of user-generated and user-related data, to make relevant recommendations and show personalized advertisements. In this case, the location is also part of the important required personal data to be shared. Last but not least, *data brokers*, e.g., Experian [exp21], TransUnion [tra21], Acxiom [acx21], etc. collect data from public and private resources, e.g., censuses, bank card transaction records, voter registration lists, etc. Most of these data are georeferenced and contain directly or indirectly location information; protecting the location of the user has become one of the most important privacy goals so far.

On the one hand, these different sources and types of data give useful feedback to the involved users and/or services, and on the other hand, when combined together, provide valuable information to various internal/external analytical services. While these activities happen within the boundaries of the law [Tan16], it is important to be able to protect the privacy (by anonymizing, perturbing, en-

This chapter was presented during the 11th International Workshop on Information Search, Integration, and Personalization [KTK16] and at the DaQuaTa International Workshop [KTK17], as well as at the São Paulo School of Advanced Science on Smart Cities [KCTK16].

crypting, etc.) the corresponding data before sharing, and to take into account the possibility of correlating, linking, and crossing diverse independent data sets. Especially the latter is becoming quite important in the era of Big Data, where the existence of diverse linked data sets is one of the promises; as an example, one can refer to the discussion on Entity Resolution problems using Linked Open Data in [ESC15]. In some cases, personal data might be so representative that even if de-identified, when integrated with a small amount of external data, one can trace back to their original source. An example case is shown in [DMHVB13], where it was discovered that four mobility traces are enough to identify 95% of the individuals in a data set. The case of location is actually one of great interest in this context, since space brings its own particular constraints. The ability to combine and correlate additional information impacts the ways we protect sensitive data and affects the privacy guarantees we can provide. Besides the explosion of online and mobile services, another important aspect is that a lot of these services actually rely on data provided by the users (*crowdsourced* data) to function, with prominent example efforts being Wikipedia [wik21], and OpenStreetMap [osm21]. Data from crowdsourced based applications, if not protected correctly, can be easily used to identify personal information, such as location or activity, and thus lead indirectly to cases of user surveillance [Lyo14].

Privacy-preserving processes usually introduce noise in the original or the aggregated data set in order to hide the sensitive information. In the case of *microdata*, a privacy-protected version, containing some synthetic data as well, is generated with the intrinsic goal to make the users indistinguishable. In the case of *statistical* data, i.e., the results of statistical queries over the original data sets,, a privacy-protected version is generated by adding noise on the actual statistical values. In both cases, we end up affecting the quality of the published data set. The privacy and the utility of the 'noisy' output are two contrasting desiderata which need to be measured and balanced. Furthermore, if we want to account for external additional information, e.g., linked or correlated data, and at the same time to ensure the same level of protection, we need to add additional noise, which inevitably deteriorates the quality of the output. This problem becomes particularly pertinent in the Big Data era, as the quality or *Veracity* is one of the five dimensions (known as the five *'V's'*) that define Big Data and where there is an abundance of external information that cannot be ignored. Since this needs to be taken into account *prior* to the publishing of the data set or the aggregated statistics there of, introducing external information into privacy-preserving techniques becomes part of the traditional processing flow while keeping an acceptable quality to privacy ratio.

As we can observe in the examples mentioned above, there are many cases where data are not protected at source (what is also described as *local* data privacy

protection) for various reasons, e.g., the users do not want to pay extra, it is impossible due to technical complexity, because the quality of the expected service will be deteriorated, etc. Thus, the burden of the privacy-preserving process falls on the various aggregators of personal/private data, who should also provide the necessary technical solutions to ensure data privacy for every user (what is also described as *global* data privacy protection).

The discussion so far explains and justifies the current situation in the privacy-preserving scientific area. As a matter of fact, a wealth of algorithms have been proposed for privacy-preserving data publishing, either for microdata or statistical data. Moreover, privacy-preserving algorithms are designed specifically for data published at one point in time (used in what we call *snapshot* data publishing) or data released over or concerning a period of time (used in what we call *continuous data publishing*). In that respect, we need to be able to correctly choose the proper privacy algorithm(s), which would allow users to share protected copies of their data with some guarantees. The selection process is far from trivial, since it is essential to:

1. select an appropriate privacy-preserving technique, relevant to the data set intended for public release;

2. understand the different requirements imposed by the selected technique and tune the different parameters according to the circumstances of the use case based on, e.g., assumptions, level of distortion, etc. [KM11];

3. get the necessary balance between privacy and data utility, which is a significant task for any privacy algorithm as well as any privacy expert.

Selecting the wrong privacy algorithm or configuring it poorly may put at risk the privacy of the involved individuals and/or end up deteriorating the quality and therefore the utility of the data set.

In data privacy research, privacy in continuous data publishing scenarios is the area that is concerned by studying the privacy problems created when sensitive data are published continuously, either infinitely, e.g., streaming data, or by multiple continuous publications over a known period of time, e.g., finite time series data. This specific subfield of data privacy becomes increasingly important since it:

(i) includes the most prominent cases, e.g., location (trajectory) privacy problems, and

(ii) provides the most challenging and yet not well charted part of the privacy algorithms since it is rather new and increasingly complex.

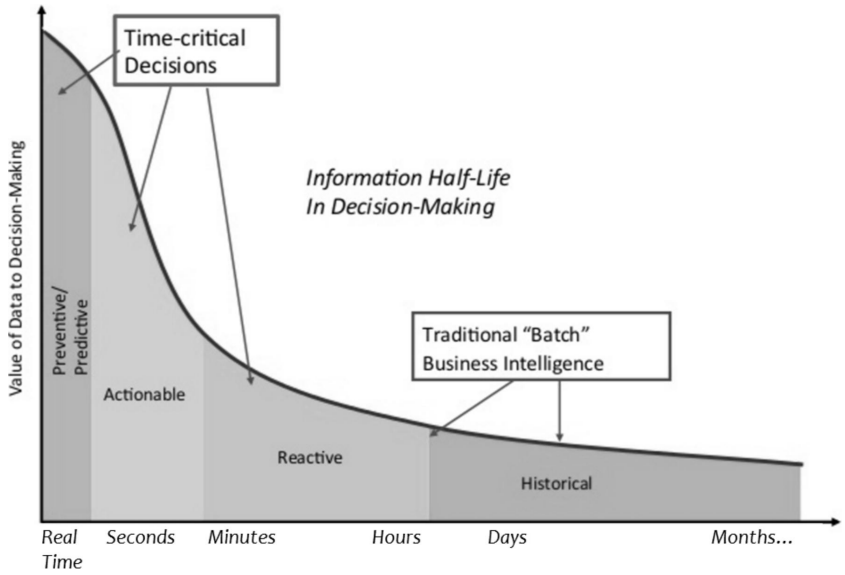


Figure 1.1: Value of data for decision-making over time from less than seconds to more than months [MGM16].

Additionally, data in continuous data publishing use cases require a timely processing because their value usually decreases over time depending on the use case as demonstrated in Figure 1.1. For this reason, we provide an insight into time-related properties of the algorithms, e.g., if they work on finite or infinite data, or if they take into consideration any underlying data dependence. The importance of continuous data publishing is stressed by the fact that, commonly, many types of data have such properties, with geospatial data being a prominent case. A few examples include—but are not limited to—data being produced while tracking the movement of individuals for various purposes (where data might also need to be privacy-protected in real-time and in a continuous fashion); crowdsourced data that are used to report measurements, such as noise or pollution (where again we have a continuous timestamped and usually georeferenced stream of data); and even isolated data items that might include location information, such as photographs or social media posts. Typically, in such cases, we have a collection of data referring to the same individual or set of individuals over a period of time, which can also be infinite. Additionally, in many cases, the privacy-preserving processes should take into account implicit correlations and restrictions that exist, e.g., space-imposed collocation or movement restrictions. Since these data are related to most of the important applications and services that enjoy high utilization rates, privacy-preserving continuous data publishing becomes one of the emblematic problems of our time.

1.1 Contribution

Our objective is to study the problems around the subject of two contrasting desiderata: quality and privacy in user-generated Big Data. We consider the scenario where data are processed/published in a continuous manner and envision a configurable privacy technique that we can tune depending on the use case requirements.

The main challenge in this setting is maintaining a balance between privacy and utility. Furthermore, we consider the presence of temporal correlation, which is inherent in continuous data publishing schemes and can lead to additional privacy loss.

Privacy, space and time The first contribution of this thesis is the survey [KTK19] of the existing literature regarding methods on privacy-preserving continuous data publishing, which appeared in the special feature on Geospatial Privacy and Security in the 19th journal of Spatial Information Science. We study works that were published over the past two decades and provide a guide that will navigate its users through the available methodology and help them select the algorithms that are fitting best their needs.

We categorize the works that we review depending on if they deal with *microdata* or *statistical data*. Then, we group them based on the duration of the processing/publishing that they aim for. Furthermore, we document in detail the privacy protection characteristics of each reviewed method.

Landmark privacy Our second contribution is the proposal and formal definition of a novel privacy notion, *landmark privacy*. Contrary to the existing privacy protection levels, our notion differentiates events between regular and events that a user might consider more privacy-sensitive, i.e. *landmarks*. The introduction of landmarks, allows for a configurable privacy protection.

First, we design and implement three landmark privacy schemes, accounting for landmarks spanning a finite time series. Thereafter, we investigate landmark privacy under temporal correlation, which is inherent in time series publishing, and study how landmarks can affect the propagation of temporal privacy loss.

Dummy landmark selection The third contribution of this thesis is the design of a module that extends our landmark privacy schemes and provides additional protection to landmarks. In other words, we answer the question *'How can we protect the fact that we care more about certain events?'*.

We design an additional differential privacy mechanism, based on the exponential mechanism, that we can easily plug in to the proposed landmark privacy

schemes. We provide an optimal solution to this problem, which we improve by adopting a heuristic approach, and then implement a more efficient module that relies on partitioning.

We extensively evaluate the methods that we propose by conducting experiments on real and synthetic data sets. We compare landmark privacy with event- and user-level privacy protection, and investigate the behavior of the overall privacy loss under temporal correlation for different distributions of landmarks. Furthermore, we estimate the impact of the privacy-preserving dummy landmark selection module on the utility of our privacy scheme.

The second and the third contributions are described in the article [KTK22], which will appear at the research papers track of the 12th ACM conference on Data and Application Security and Privacy.

1.2 Structure

This thesis is structured as follows:

Chapter 2 introduces some relevant terminology and information around the problem of quality and privacy in user-generated Big Data with a special focus on continuous data publishing. First, in Section 2.1, we categorize user-generated data sets and review data processing in the context of continuous data publishing. Second, in Section 2.2, we define information disclosure in data privacy. We list the categories of privacy attacks, the possible privacy protection levels, the fundamental privacy operations that are applied to achieve data privacy, and finally we provide a brief overview of the basic notions for data privacy protection. Third, in Section 2.3, we focus on the impact of correlation on data privacy. More particularly, we discuss the different types of correlation, we document ways to extract data correlation from continuous data, and we investigate the privacy risks that data correlation entails with special focus on the privacy loss under temporal correlation.

Chapter 3 reviews works that deal with privacy under continuous data publishing covering diverse use cases. We present the relevant literature based on two levels of categorization. First, we group works with respect to whether they deal with microdata or statistical data as input. Then, we further group them into two subcategories depending on if they are designed for the finite or infinite observation setting.

Chapter 4 puts forward a novel configurable privacy scheme, *landmark privacy* (Section 4.1), which takes into account significant events (*landmarks*) in the time series and allocates the available privacy budget accordingly. We propose three privacy schemes that guarantee landmark privacy. To further enhance our privacy methodology, and protect the landmark position in the time series, we propose techniques to perturb the initial landmark set (Section 4.2).

Chapter 5 presents the experiments that we performed in order to evaluate landmark privacy (Chapter 4) on real and synthetic data sets. Section 5.1 contains all the details regarding the data sets the we used for our experiments along with the system configurations. Section 5.2 evaluates the data utility of the landmark privacy schemes that we designed in Section 4.1 and investigates the behavior of the privacy loss under temporal correlation for different distributions of landmarks. Section 5.3 justifies our decisions while designing the privacy-preserving landmark selection module in Section 4.2 and the data utility impact of the latter. Finally, Section 5.4 concludes this chapter by summarizing the main results derived from the experiments.

Chapter 6 concludes the thesis and outlines possible future directions.

Chapter 2

Preliminaries

In this chapter, we introduce some relevant terminology and information around the problem of quality and privacy in user-generated Big Data with a special focus on continuous data publishing. First, in Section 2.1, we categorize user-generated data sets, that we consider in a tabular form, and review data processing in the context of continuous data publishing. Second, in Section 2.2, we define information disclosure in data privacy. Thereafter, we list the categories of privacy attacks, the possible privacy protection levels, the fundamental privacy operations that are applied to achieve data privacy, and finally we provide a brief overview of the basic notions for data privacy protection. Third, in Section 2.3, we focus on the impact of correlation on data privacy. More particularly, we discuss the different types of correlation, we document ways to extract data correlation from continuous data, and we investigate the privacy risks that data correlation entails with special focus on the privacy loss under temporal correlation.

2.1 Data sets and data publishing

In this section, we categorize user-generated data sets in terms of their form and their processing and publishing.

2.1.1 Data categories

In this thesis, we are interested in data that contain information about individuals and their actions, as these are highly privacy-sensitive. A typical category of such data are *user-generated data* which are the outcome of users–services interactions, e.g., social media, location-based services (LBS), etc. These interactions result in the generation of *data items* which are tuples that typically contain a user identifier, a timestamp, and context information (e.g., location, activity, etc.) We

Name	Age	Location	Status
Donald	27	Le Marais	at work
Daisy	25	Belleville	driving
Huey	12	Montmartre	running
Dewey	11	Montmartre	at home
Louie	10	Quartier Latin	walking
Quackmore	62	Opéra	dining

(a) Microdata

Location	Count
Belleville	1
Quartier Latin	1
Le Marais	1
Montmartre	2
Opéra	1

(b) Statistical data

Figure 2.1: Example of raw user-generated (a) microdata, and the related (b) statistical data for a specific timestamp.

firstly classify data based on their form in:

- *Microdata* (Figure 2.1a) are the data items in their raw, usually tabular, form pertaining to individuals.
- *Statistical data* (Figure 2.1b) are the outcome of statistical processes on microdata, e.g., average, count, sum, etc.

To accompany and facilitate the descriptions in this chapter, we provide Example 2.1.1 as a running example.

Example 2.1.1. *Users interact with an LBS by making queries in order to retrieve some useful location-based information or just reporting user-state at various locations. This user–LBS interaction generates user-related data, organized in a schema with the following attributes:* Name *(the unique identifier of the table),* Age*,* Location*, and* Status *(Figure 2.1a). The 'Status' attribute includes information that characterizes the user state or the query itself, and its value varies according to the service functionality. Subsequently, the generated data are aggregated (by issuing count queries over them) in order to derive useful information about the popularity of the venues during the day (Figure 2.1b).*

An example of microdata is displayed in Figure 2.1a, while an example of statistical data in Figure 2.1b. Data, in either of these two forms, may have a special property called *continuity*, i.e., their values change and can be observed through time. Observing the evolution of the data attribute values over time may offer valuable insight regarding the underlying population not only about the past but also both about the present and future. Depending on the span of the observation, we categorize data in:

Name	Age	Location	Status
Donald	27	Le Marais	at work
Daisy	25	Belleville	driving
Huey	12	Montmartre	running
Dewey	11	Montmartre	at home
Louie	10	Quartier Latin	walking
Quackmore	62	Opéra	dining

t_1

Name	Age	Location	Status
Donald	27	Montmartre	driving
Daisy	25	Montmartre	at the mall
Huey	12	Quartier Latin	sightseeing
Dewey	11	Opéra	walking
Louie	10	Quartier Latin	at home
Quackmore	62	Montmartre	biking

t_2 ...

(a) Microdata

Location	Count		
	t_1	t_2	...
Belleville	1	0	...
Quartier Latin	1	2	...
Le Marais	1	0	...
Montmartre	2	3	...
Opéra	1	1	...

(b) Statistical data

Figure 2.2: (a) Microdata, and (b) the corresponding statistics at multiple timestamps.

- *Finite data* are observed during a predefined time interval.
- *Infinite data* are observed in an uninterrupted fashion.

Example 2.1.2. *Extending Example 2.1.1, Figure 2.2 shows an example of continuous data, by introducing one data table for each consecutive timestamp. The two data tables over the time span* $[t_1, t_2]$ *are an example of finite data. Infinite data are the whole series of data obtained over the period* $[t_1, \infty)$ *(infinity is denoted by '...').*

We further define two sub-categories, which are not exhaustive, i.e., not all data sets belong to the one or the other category, applicable to both finite and infinite data:

- *Sequential data* have variable values that change depending on their previous values. For example, trajectories are finite sequences of location stamps, as naturally the position at each timestamp is connected to the position at the previous timestamp.

- *Incremental data* are augmented at each subsequent timestamp with supplementary information. For example, trajectories can be considered as incremental data when at each timestamp we consider all the previously visited locations by an individual incremented by the individual's current position.

2.1.2 Data processing and publishing

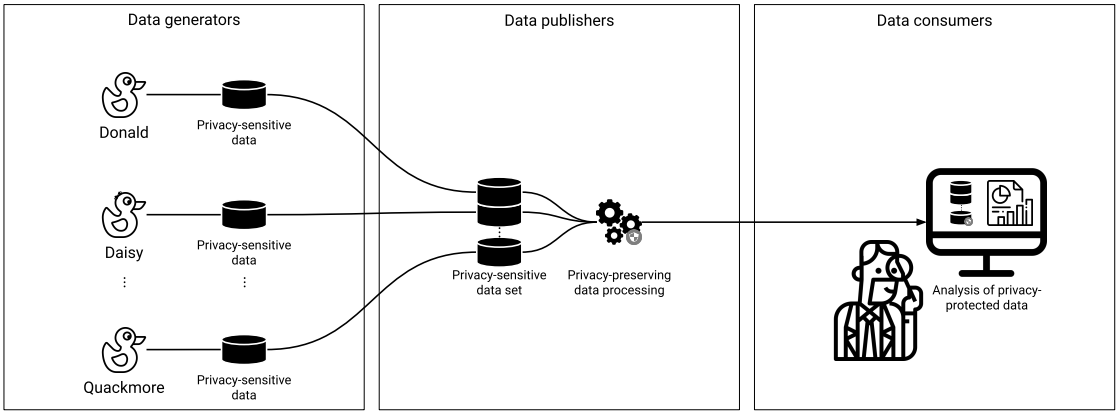
We categorize data processing and publishing based on what entity has access to the raw data in the following schemes:

- *Global scheme* (Figure 2.3a) dictates the collection, processing and privacy-protection, and then publishing of the data by a central (trusted) entity, e.g., [McS09, BBDS13, JNS18].

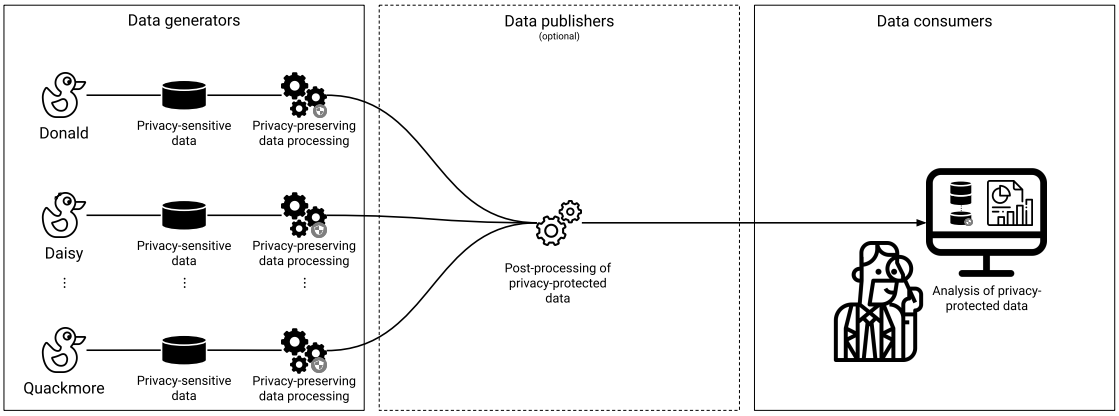
- *Local scheme* (Figure 2.3b) requires the storage, processing and privacy-protection of data on the side of data generators before sending them to any intermediate or final entity, e.g., [ABCP13, EPK14, KLPT17].

In the case of location data privacy, data processing and publishing methods are divided in *service-* and *data*-centric [CM11]. The service-centric methods correspond to scenarios where individuals share their privacy-protected location with a service to get some relevant information (local publishing scheme). The data-centric methods relate to the publishing of user-generated data to data consumers (global publishing scheme).

There is a long-standing debate whether the local or the global architectural scheme is more efficient with respect to not only privacy, but also organizational, economic, and security factors [Kin83]. On the one hand, in the global privacy scheme (Figure 2.3a), the dependence on third-party entities poses the risk of arbitrary privacy leakage from a compromised data publisher. Nonetheless, the expertise of these entities is usually superior to that of the majority of (non-technical) data generators' in terms of understanding privacy permissions/policies and setting-up relevant preferences. Moreover, in the global architecture, less distortion is necessary before publicly releasing the aggregated data set, naturally because the data sets are larger and users can be 'hidden' more easily. On the other hand, the local privacy scheme (Figure 2.3b) facilitates fine-grained data management, offering to every individual better control over their data [Gol98]. Nonetheless, data distortion at an early stage might prove detrimental to the overall utility of the aggregated data set. The so far consensus is that there is no overall optimal solution among the two designs. Most service-providing companies prefer the global scheme, mainly for reasons of better management and control over the data, while several privacy advocates support the local privacy scheme



(a) Global scheme



(b) Local scheme

Figure 2.3: The usual flow of user-generated data, optionally harvested by data publishers, privacy-protected, and released to data consumers, according to the (a) global, and (b) local privacy schemes.

that offers users full control over what and how data are published. Although there have been attempts to bridge the gap between them, e.g., [BEM+17], the global scheme is considerably better explored and implemented [Sat17].

We distinguish publishing modes for private data between:

- *Snapshot mode* (also appearing as *one-shot* or *one-off* publishing) processes and releases a data set at a specific point in time and thereafter is not concerned anymore with the specific data set. For example, in Figure 2.4a (ignore the privacy-preserving step for the moment) individuals send their data to an LBS provider, considering a specific timestamp. The use cases of continuous data publishing abound, with the proliferation of the Internet, sensors, and connected devices, which produce and send to servers huge amounts of continuous personal data in astounding speed.

- *Continuous mode* computes and publishes augmented or updated versions of one data set in different timestamps, and without a predefined duration. In the context of privacy-preserving data publishing, privacy preservation is tightly coupled with the data processing and publishing stages.

  We further categorize continuous publishing mode into:

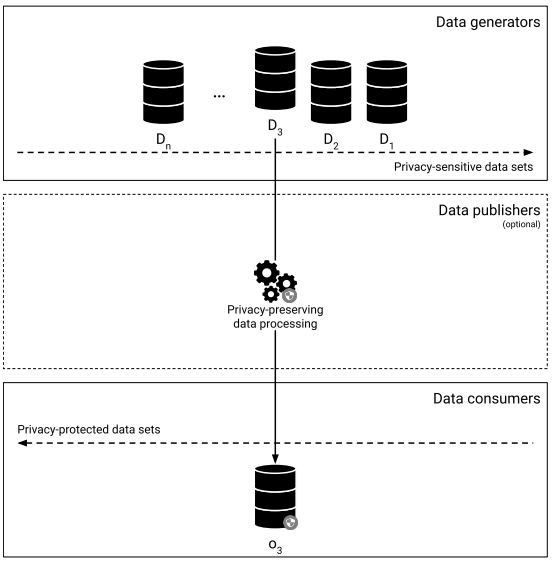
  - *Batch mode* (Figure 2.4b) considers data in groups in specific time intervals. It is performed (usually offline) over both finite and infinite data
  - *Streaming mode* (Figure 2.4c) processes data per timestamp, infinitely. It is by definition connected to infinite data (usually in real-time).

2.2 Data privacy

In this section we first study the notion of information disclosure and focus on the privacy attacks that can lead to it. Furthermore, we investigate the possible privacy protection levels in continuous data publishing. Finally, we identify the most common privacy operations and the seminal works for privacy-preserving data publishing.

2.2.1 Information disclosure

When personal data are publicly released, either as microdata or statistical data, individuals' privacy can be compromised, i.e, an adversary becomes certain about an individual's *sensitive attribute*, i.e., personal information, with a probability



(a) Snapshot mode

(b) Batch mode (c) Streaming mode

Figure 2.4: The different data processing and publishing modes of continuously generated data sets. (a) Snapshot publishing, (b) continuous publishing–batch mode, and (c) continuous publishing–streaming mode. $\boldsymbol{o}_x$ denotes the privacy-protected version of the data set D_x or statistics thereof, while '...' denote the continuous data generation and/or publishing, where applicable. Depending on the data observation span, n can either be finite or tend to infinity.

higher than a desired threshold. In the literature, this incident is known as *information disclosure* and is usually categorized [LLV07, WCFY10, NS08] as:

- *Presence disclosure* takes place when the participation or absence of an individual in a data set is revealed.
- *Identity disclosure* links an individual to a particular record.
- *Attribute disclosure* reveals information (attribute value) about an individual.

In the literature, identity disclosure is also referred to as *record linkage*, and presence disclosure as *table linkage*. Notice that identity disclosure can result in attribute disclosure, and vice versa.

To better illustrate these definitions, we provide some examples based on Figure 2.1. Presence disclosure appears when by looking at the (privacy-protected) counts of Figure 2.1b, we can guess if Quackmore has participated in Figure 2.1a. Identity disclosure appears when we can guess that the sixth record of (a privacy-protected version of) the microdata of Figure 2.1a belongs to Quackmore. Attribute disclosure appears when it is revealed from (a privacy-protected version of) the microdata of Figure 2.1a that Quackmore is 62 years old.

2.2.2 Attacks to privacy

Information disclosure is typically achieved by combining supplementary (background) knowledge with the released data or by setting unrealistic assumptions while designing the privacy-preserving algorithms. In its general form, this is known as *adversarial* or *linkage* attack. Even though many works directly refer to the general category of linkage attacks, we distinguish also the following sub-categories:

- *Sensitive attribute domain knowledge* can result in *homogeneity and skewness* attacks [MGKV06, LLV07], when statistics of the sensitive attribute values are available, and *similarity attack*, when semantics of the sensitive attribute values are available.
- *Complementary release attacks* [Swe02b] take place when attackers take into account previous releases of different versions of the same and/or related data sets. In this category, we also identify the *unsorted matching* attack [Swe02b], which is achieved when two privacy-protected versions of an original data set are published in the same tuple ordering. Other instances include: (i) the *join* attack [WF06], when tuples can be identified by joining (on the non

uniquely identifying attributes, i.e., *quasi-identifiers*) several releases, (ii) the *tuple correspondence* attack [FWFP08], when in case of incremental data certain tuples correspond to certain tuples in other releases, in an injective way, (iii) the *tuple equivalence* attack [HBN11], when tuples among different releases are found to be equivalent with respect to the sensitive attribute, and (iv) the *unknown releases* attack [ST15], when the privacy preservation is performed without taking into account previous data releases.

- *Data correlation* that may exist either within one data set or among one data set and previous data releases, and/or other external sources [KM11, CFYD14, LCM16, ZZP17]. We will look into this category in more detail later in Section 2.3.

The first sub-category of attacks has been mainly addressed in works on snapshot microdata publishing, but is also present in continuous publishing; however, algorithms for continuous publishing typically accept the proposed solutions for the snapshot publishing scheme (see discussion over k-anonymity and l-diversity in Section 2.2.5). This kind of attacks is tightly coupled with publishing the (privacy-protected) sensitive attribute value. An example is the lack of diversity in the sensitive attribute domain, e.g., if all users in the data set of Figure 2.1a had *running* as their Status (the sensitive attribute). The second and third subcategories are attacks emerging (mostly) in continuous publishing scenarios. Consider again the data set in Figure 2.1a. The complementary release attack means that an adversary can learn more things about the individuals (e.g., that there are high chances that Donald was at work) if he/she combines the information of two privacy-protected versions of this data set. By the data correlation attack, the status of Donald could be more certainly inferred, by taking into account the status of Dewey at the same moment and the dependencies between Donald's and Dewey's status, e.g., when Dewey is at home, then most probably Donald is at work. In order to better protect the privacy of Donald in case of attacks, the data should be privacy-protected in a more adequate way (than without the attacks).

2.2.3 Levels of privacy protection

In continuous data publishing we consider the privacy protection level with respect to not only the users, but also to the *events* occurring in the data. An event is a pair of an identifying attribute of an individual and the sensitive data (including contextual information) and we can see it as a correspondence to a record in a database, where each individual may participate once. Data publishers typically release events in the form of sequences of data items, usually indexed in time order (time series) and geotagged, e.g., ('Dewey', 'at home at Montmartre at t_1'), ...,

('Quackmore', 'dining at Opéra at t_1'). We use the term 'users' to refer to the *individuals*, also known as *participants*, who are the source of the processed and published data. Therefore, they should not be confused with the consumers of the released data sets. Users are subject to privacy attacks, and thus are the main point of interest of privacy protection mechanisms. The possible privacy protection levels are:

(a) *Event-level* [DNPR10, DNP$^+$10] limits the privacy protection to *any single event* in a time series, providing high data utility.

(b) *User-level* [DNPR10, DNP$^+$10] protects *all the events* in a time series, providing high user privacy.

(c) *w-event-level* [KPXP14] provides privacy protection to *any sequence of w events* in a time series. privacy protection.

Figure 2.5 demonstrates the application of the possible protection levels on the statistical data of Example 2.1.2. For instance, in event-level (Figure 2.5a) it is hard to determine whether Quackmore was dining at Opéra at t_1. Moreover, in user-level (Figure 2.5b) it is hard to determine whether Quackmore was ever included in the released series of events at all. Finally, in 2-event-level (Figure 2.5c) it is hard to determine whether Quackmore was ever included in the released series of events between the timestamps t_1 and t_2, t_2 and t_3, etc. (i.e., for a window $w = 2$).

Location	Count		
	t_1	t_2	...
Belleville	1	0	...
Quartier Latin	1	2	...
Le Marais	1	0	...
Montmartre	2	3	...
Opéra	1	1	...

(a) Event-level

Location	Count		
	t_1	t_2	...
Belleville	1	0	...
Quartier Latin	1	2	...
Le Marais	1	0	...
Montmartre	2	3	...
Opéra	1	1	...

(b) User-level

Location	Count		
	t_1	t_2	...
Belleville	1	0	...
Quartier Latin	1	2	...
Le Marais	1	0	...
Montmartre	2	3	...
Opéra	1	1	...

(c) 2-event-level

Figure 2.5: Protecting the data of Figure 2.2b on (a) event-, (b) user-, and (c) 2-event-level. A suitable distortion method can be applied accordingly.

Contrary to event-level, which provides privacy guarantees for a single event, user- and w-event-level offer stronger privacy protection by protecting a series of events. Event- and w-event-level better fit scenarios of infinite data observation, whereas user-level is more appropriate when the span of data observation is finite. w-event- is narrower than user-level protection due to its sliding window processing

methodology. In the extreme cases where w is equal either to 1 or to the length of the time series, w-event- matches event- or user-level protection, respectively. Although the described levels have been coined in the context of *differential privacy* [DMNS06], a seminal privacy method that we will discuss in more detail in Section 2.2.5, they are used for other privacy protection techniques as well.

2.2.4 Privacy-preserving operations

We identify the following privacy operations that can be applied on the original data to achieve privacy preservation:

- *Aggregation* combines multiple rows of a data set to form a single value which will replace these rows.
- *Generalization* replaces an attribute value with a parent value in the attribute taxonomy (when applicable).
- *Suppression* deletes completely certain sensitive values or entire records.
- *Perturbation* disturbs the initial attribute value in a deterministic or probabilistic way. The probabilistic data distortion is referred to as *randomization*.

For example, consider the table schema *User(Name, Age, Location, Status)*. If we want to protect the *Age* of the user by aggregation, we may group the data by Location and report the average Age for each group; by generalization, we may replace the Age by Age intervals; by suppression we may delete the entire table column corresponding to Age; by perturbation, we may augment each Age by a predefined percentage of the Age; by randomization we may randomly replace each Age by a value taken from the probability density function of the attribute.

It is worth mentioning that there is a series of algorithms (e.g., [BCHL09, KL10, CWL+14]) based on the *cryptography* operation. However, the majority of these methods, among other assumptions that they make, have minimum or even no trust to the entities that handle the personal information. Furthermore, the amount and the way of data processing of these techniques usually burden the overall procedure, deteriorate the utility of the resulting data sets to a point where they are completely useless, and thus restrict their usage by third-parties. Our focus is limited to techniques that achieve a satisfying balance between both participants' privacy and data utility.

2.2.5 Basic notions for privacy protection

For completeness, in this section we present the seminal works for privacy-preserving data publishing, which, even though originally designed for the

snapshot publishing scenario, have paved the way of privacy-preserving continuous publishing as well.

Microdata

Sweeney coined *k-anonymity* [Swe02b], one of the first established works on data privacy. A released data set features k-anonymity protection when the values of a set of identifying attributes, called the *quasi-identifiers*, is the same for at least k records in the data set. In a follow-up work [Swe02a], the author describes a way to achieve k-anonymity for a data set by the suppression or generalization of certain values of the quasi-identifiers.

Several works identified and addressed privacy concerns on k-anonymity. Machanavajjhala et al. [MGKV06] pointed out that k-anonymity is vulnerable to homogeneity and background knowledge attacks. Thereby, they proposed *l-diversity*, which demands that the values of the sensitive attributes are 'well-represented' by l sensitive values in each group. Principally, a data set can be l-diverse by featuring at least l distinct values for the sensitive field in each group (*distinct* l-diversity). Other instantiations demand that the entropy of the whole data set is greater than or equal to $\log(l)$ (*entropy* l-diversity) or that the number of appearances of the most common sensitive value is less than the sum of the counts of the rest of the values multiplied by a user defined constant c (*recursive (c, l)*-diversity). Later on, Li et al. [LLV07] indicated that l-diversity can be void by skewness and similarity attacks due to sensitive attributes with a small value range. In such cases, *θ-closeness* guarantees that the distribution of a sensitive attribute in a group and the distribution of the same attribute in the whole data set is 'similar'. This similarity is bound by a threshold θ. A data set features θ-closeness when all of its groups satisfy θ-closeness.

The main drawback of k-anonymity (and its derivatives) is that it is not tolerant to external attacks of re-identification on the released data set. The problems identified in [Swe02b] appear when attempting to apply k-anonymity on continuous data publishing (as we will also see next in Section 3.1). These attacks include multiple k-anonymous data set releases with the same record order, subsequent releases of a data set without taking into account previous k-anonymous releases, and tuple updates. Proposed solutions include rearranging the attributes, setting the whole attribute set of previously released data sets as quasi-identifiers or releasing data based on previous k-anonymous releases [SNE17].

Statistical data

While methods based on k-anonymity have been mainly employed for releasing microdata, *differential privacy* [DMNS06] has been proposed for releasing high

utility aggregates over microdata while providing semantic privacy guarantees that characterize the output data. Differential privacy is algorithmic, it characterizes the data publishing process, which passes its privacy guarantee to the resulting data. It ensures that any adversary observing a privacy-protected output, no matter their computational power or auxiliary information, cannot conclude with absolute certainty if an individual is included in the input data set (Definition 1).

Definition 1 (Neighboring data sets [DMNS06]). *Two data sets are neighboring (or adjacent) when they differ by at most one tuple, i.e., one can be obtained by adding/removing the data of an individual to/from the other.*

Moreover, differential privacy quantifies and bounds the impact that the addition/removal of an individual to/from a data set has on the derived privacy-protected aggregates thereof. More precisely, differential privacy quantifies the impact of the addition/removal of a single tuple in D on the output $\boldsymbol{o}$ of a privacy mechanism $\mathcal{M}$ that perturbs the result of a query function f. The distribution of all $\boldsymbol{o}$, in some range $\mathcal{O}$, is not affected *substantially*, i.e., it changes only slightly due to the modification of any one tuple in all possible $D \in \mathcal{D}$. Formally, differential privacy is given in Definition 2.

Definition 2 (Differential privacy [DMNS06]). *A privacy mechanism $\mathcal{M}$, with domain $\mathcal{D}$ and range $\mathcal{O}$, satisfies ε-differential privacy, for a given privacy budget ε, if for every pair of neighboring data sets $D, D' \in \mathcal{D}$ and all sets $O \subseteq \mathcal{O}$:*

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in O]$$

$\Pr[\cdot]$ denotes the probability of $\mathcal{M}$ generating an output from $O \subseteq \mathcal{O}$, when given D as input. The *privacy budget* ε is a positive real number that represents the user-defined privacy goal [McS09]. As the definition implies, $\mathcal{M}$ achieves stronger privacy protection for lower values of ε since the probabilities of D and D' being true worlds are similar, but the utility of the output is reduced since more randomness is introduced by $\mathcal{M}$. The privacy budget ε is usually set to 0.01, 0.1, or, in some cases, $\ln 2$ or $\ln 3$ [LC11].

The applicability of differential privacy mechanisms is inseparable from the query's function sensitivity. The presence/absence of a single record should only change the result slightly, and therefore differential privacy methods are best for low sensitivity queries (see Definition 3) such as counts. However, sum, max, and in some cases average queries can be problematic, since a single, outlier value could change the output noticeably, making it necessary to add a lot of noise to the query's answer.

Definition 3 (Query function sensitivity [DMNS06]). *The sensitivity of a query function f for all neighboring data sets $D, D' \in \mathcal{D}$ is:*

$$\Delta f = \max_{D, D' \in \mathcal{D}} \|f(D) - f(D')\|_1$$

The notion of differential privacy has highly influenced the research community, resulting in many follow-up publications ([MT07, KM11, ZCP+17] to mention a few). We distinguish here *Pufferfish* [KM14]. *Pufferfish* is a framework that allows experts in an application domain, without necessarily having any particular expertise in privacy, to develop privacy definitions for their data sharing needs. To define a privacy mechanism using *Pufferfish*, one has to define a set of potential secrets $\mathcal{X}$, a set of distinct pairs $\mathcal{X}_{pairs}$, and auxiliary information about data evolution scenarios $\mathcal{B}$. $\mathcal{X}$ serves as an explicit specification of what we would like to protect, e.g., 'the record of an individual x is (not) in the data'. $\mathcal{X}_{pairs}$ is a subset of $\mathcal{X} \times \mathcal{X}$ that instructs how to protect the potential secrets $\mathcal{X}$, e.g., ('x is in the table', 'x is not in the table'). Finally, $\mathcal{B}$ is a set of conservative assumptions about how the data evolved (or were generated) that reflects the adversary's belief about the data, e.g., probability distributions, variable correlation, etc. When there is independence between all the records in the original data set, then ε-differential privacy and the privacy definition of ε-*Pufferfish*$(\mathcal{X}, \mathcal{X}_{pairs}, \mathcal{B})$ are equivalent.

Popular privacy mechanisms A typical example of a differential privacy mechanism is the *Laplace mechanism* [DR+14]. It draws randomly a value from the probability distribution of Laplace(μ, b), where μ stands for the location parameter and $b > 0$ is the scale parameter (Figure 2.6). In our case, μ is equal to the original output value of a query function, and b is the sensitivity of the query function divided by the privacy budget ε. The Laplace mechanism works for any function with range the set of real numbers.

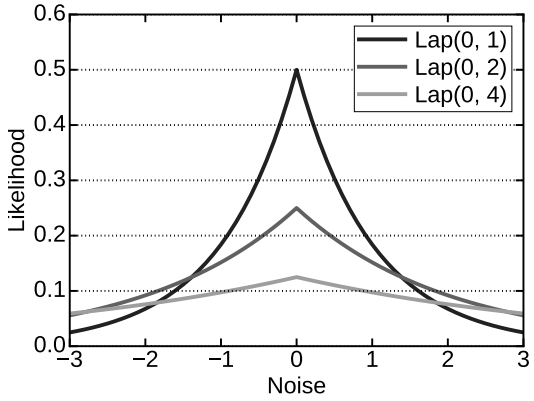


Figure 2.6: A Laplace distribution for location $\mu = 0$ and different scale values b.

A specialization of this mechanism for location data is the *Planar Laplace mechanism* [ABCP13, CPS15], an adaptation of differential privacy for location data in snapshot publishing (*Geo-indistinguishability*). It is based on l-privacy, which of-

fers to individuals within an area with radius r a privacy level of l (Figure 2.7). More specifically, l is equal to εr if any two locations within distance r provide data with similar distributions. This similarity depends on r because the closer two locations are, the more likely they are to share the same features. Intuitively, the definition implies that if an adversary learns the published location for an individual, the adversary cannot infer the individual's true location, out of all the points in a radius r, with a certainty higher than a factor depending on l. The technique adds random noise drawn from a multivariate Laplace distribution to individuals' locations, while taking into account spatial boundaries and features.

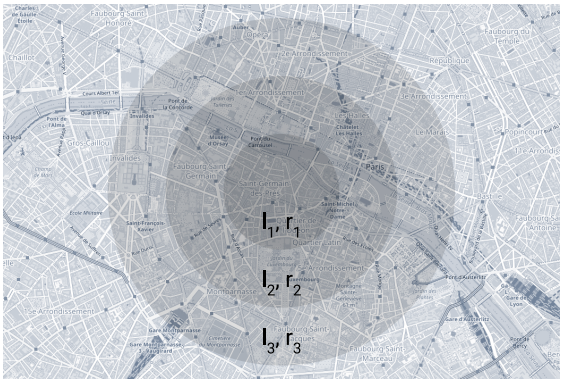


Figure 2.7: Geo-indistinguishability: privacy level l varying with the protection radius r.

For query functions that do not return a real number, e.g., 'What is the most visited country this year?', or in cases where perturbing the value of the output will completely destroy its utility, e.g., 'How many patients in the ICU?', most works in the literature use the *Exponential mechanism* [MT07]. Initially, a utility function u, with sensitivity Δu, maps pairs of the input value x and output value r to utility scores. Thereafter, the mechanism M selects an output value r from a set of possible outputs R with probability proportional to $\exp(\frac{\varepsilon u(x,r)}{2\Delta u})$.

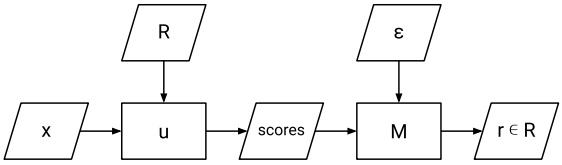


Figure 2.8: The internal mechanics of the exponential mechanism.

Another technique for differential privacy mechanisms is the *randomized response* [War65]. It is a privacy-preserving survey method that introduces probabilistic noise to the statistics of a research by randomly instructing respondents to

answer truthfully or 'Yes' to a sensitive, binary question. The technique achieves this randomization by including a random event, e.g., the flip of a fair coin. The respondents reveal to the interviewers only their answer to the question, and keep as a secret the result of the random event (i.e., if the coin was tails or heads). Thereafter, the interviewers can calculate the probability distribution of the random event, e.g., $\frac{1}{2}$ heads and $\frac{1}{2}$ tails, and thus they can roughly eliminate the false responses and estimate the final result of the research. Based on this methodology, the *Random response* mechanism [WCFY10] returns the true or flipped answer value x with a probability p proportional to the privacy budget ε (Figure 2.9).

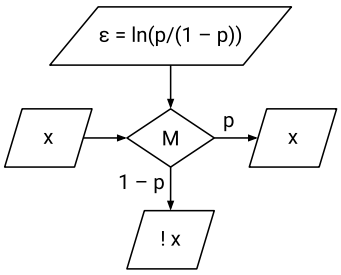


Figure 2.9: The internal mechanics of the random response mechanism.

A special category of differential privacy-preserving mechanisms is that of *pan-private* algorithms [DNP+10]. Pan-private algorithms hold their privacy guarantees even when snapshots of their internal state (memory) are accessed during their execution by an external entity, e.g., subpena, security breach, etc. There are two intrusion types that a data publisher has to deal with when designing a pan-private mechanism: *single unannounced*, and *continual announced* intrusion. In the first, the data publisher assumes that the mechanism's state is observed by the external entity one unique time, without the data publisher ever being notified about it. In the latter, the external entity gains access to the mechanism's state multiple times, and the publisher is notified after each time. The simplest approach to deal with both cases is to make sure that the data in the memory of the mechanism have constantly the same distribution, i.e., they are differentially private. Notice that this must hold throughout the mechanism's lifetime, even before/after it processes any sensitive data item(s).

In what follows, we present some primordial properties of differential private mechanisms that rule their composition and post processing.

Composition Mechanisms that satisfy differential privacy are *composable*, i.e., the combination of their results satisfy differential privacy as well. In this section, we provide an overview of the most prominent composition theorems

that instruct data publishers *how* to estimate the overall privacy protection when utilizing a series of differential privacy mechanisms.

Theorem 1 (Composition [McS09]). *Any combination of a set of independent differential privacy mechanisms satisfying a corresponding set of privacy guarantees shall satisfy differential privacy as well, i.e., provide a differentially private output.*

Generally, when we apply a series of independent (i.e., in the way that they inject noise) differential privacy mechanisms on independent data, we can calculate the privacy level of the resulting output according to the *sequential* composition property [McS09, SCDF16].

Theorem 2 (Sequential composition on independent data [McS09]). *The privacy guarantee of $m \in \mathbb{Z}^+$ independent privacy mechanisms, satisfying ε_1-, ε_2-, ..., ε_m-differential privacy respectively, when applied over the same data set equals to $\sum_{i=1}^{m} \varepsilon_i$.*

Asking a series of queries may allow the disambiguation between possible data sets, making it necessary to add even more noise to the outputs. Keeping the original guarantee across multiple queries that require different/new answers requires the injection of noise proportional to the number of the executed queries, and thus destroying the utility of the output. For this reason, after a series of queries exhausts the available privacy budget the data set has to be discarded.

Notice that the sequential composition corresponds to the worst case scenario where each time we use a mechanism we have to invest some (or all) of the available privacy budget. In the special case that we query disjoint data sets, we can take advantage of the *parallel* composition property [McS09, SCDF16], and thus spare some of the available privacy budget.

Theorem 3 (Parallel composition on independent data [McS09]). *When $m \in \mathbb{Z}^+$ independent privacy mechanisms, satisfying ε_1-, ε_2-,..., ε_m-differential privacy respectively, are applied over disjoint independent subsets of a data set, they provide a privacy guarantee equal to $\max_{i\in[1,m]} \varepsilon_i$.*

When the users consider recent data releases more privacy-sensitive than distant ones, we estimate the overall privacy loss in a time fading manner according to a temporal discounting function, e.g., exponential, hyperbolic, [Far20].

Theorem 4 (Sequential composition with temporal discounting [Far20]). *A set of $m \in \mathbb{Z}^+$ independent privacy mechanisms, satisfying ε_1-, ε_2-,..., ε_m-differential privacy respectively, satisfy $\sum_{i=1}^{m} g(i)\varepsilon_i$ differential privacy for a discount function g.*

When dealing with temporally correlated data, we handle a sequence of $w \leq t \in \mathbb{Z}^+$ mechanisms (indexed by $m \in [1, t]$) as a single entity where each mechanism contributes to the temporal privacy loss depending on its order of application [CYXX17]. The first ($m-1$ if $w \leq 2$ or $m-w+1$ if $w > 2$) and last (m) mechanisms contribute to the backward and forward temporal privacy loss respectively (see also Section 2.3.4). When w is greater than 2, the rest of the mechanisms (between $m-w+2$ and $m-1$) contribute only to the privacy loss that is corresponding to the publication of the relevant data.

Theorem 5 (Sequential composition under temporal correlation [CYXX18])**.** *When a set of $w \leq t \in \mathbb{Z}^+$ independent privacy mechanisms, satisfying $\varepsilon_{m\in[1,t]}$-differential privacy, is applied over a sequence of an equal number of temporally correlated data sets, it provides a privacy guarantee equal to:*

$$\begin{cases} \alpha^B_{m-1} + \alpha^F_m & w \leq 2 \\ \alpha^B_{m-w+1} + \alpha^F_m + \sum_{i=m-w+2}^{m-1} \varepsilon_i & w > 2 \end{cases}$$

Notice that the estimation of forward privacy loss is only pertinent to a setting under finite observation and moderate correlation. In different circumstances, it might be impossible to calculate the upper bound of the temporal privacy loss, and thus only the backward privacy loss would be relevant.

Post-processing Every time a data publisher interacts with (any part of) the original data set, it is mandatory to consume some of the available privacy budget according to the composition theorems 2 and 3. However, the *post-processing* of a perturbed data set can be done without using any additional privacy budget.

Theorem 6 (Post-processing [McS09])**.** *The post-processing of any output of an ε-differential privacy mechanism shall not deteriorate its privacy guarantee.*

Naturally, using the same (or different) privacy mechanism(s) multiple times to interact with raw data in combination with already perturbed data implies that the privacy guarantee of the final output will be calculated according to Theorem 2. That is, we add up the privacy budgets attributed to the outputs from previous mechanism applications with the current privacy budget.

Example 2.2.1. *To illustrate the usage of the microdata and statistical data techniques for privacy-preserving data publishing, we revisit Example 2.1.2. In this example, users continuously interact with an LBS by reporting their status at various locations. Then, the reported data are collected by the central service, in order to be protected and then published, either as a whole, or as statistics thereof. Notice that in order to showcase the straightforward application of k-anonymity and differential privacy, we apply the two methods on each timestamp independently from*

Name	Age	Location	Status
*	> 20	Paris	at work
*	> 20	Paris	driving
*	> 20	Paris	dining
*	≤ 20	Paris	running
*	≤ 20	Paris	at home
*	≤ 20	Paris	walking

t_1

Name	Age	Location	Status
*	> 20	Paris	driving
*	> 20	Paris	at the mall
*	> 20	Paris	biking
*	≤ 20	Paris	sightseeing
*	≤ 20	Paris	walking
*	≤ 20	Paris	at home

t_2 ...

Figure 2.10: 3-anonymous event-level protected versions of the microdata in Table 2.2a.

Location	Count
Belleville	1
Quartier Latin	1
Le Marais	1
Montmartre	2
Opéra	1

(a) True counts

$\xrightarrow{\text{Noise}}$

Location	Count
Belleville	1
Quartier Latin	0
Le Marais	2
Montmartre	3
Opéra	1

(b) Perturbed counts

Figure 2.11: (a) The original version of the data of Figure 2.2b, and (b) their 1-differentially event-level private version.

the previous one, and do not take into account any additional threats imposed by continuity.

First, we anonymize the data set of Figure 2.2a using k-anonymity, with $k = 3$*. This means that any user should not be distinguished from at least* 2 *others. Status is the sensitive attribute, thus the attribute that we wish to protect. We start by suppressing the values of the Name attribute, which is the identifier. The Age and Location attributes are the quasi-identifiers, so we proceed to adequately generalize them. We turn age values to ranges (*≤ 20*, and* > 20*), and generalize location to city level (Paris). Finally, we achieve* 3*-anonymity by putting the entries in groups of three, according to the quasi-identifiers. Figure 2.10 depicts the results at each timestamp.*

Next, we demonstrate differential privacy. We apply an ε*-differentially private Laplace mechanism, with* $\varepsilon = 1$*, taking into account the count query that generated the true counts of Figure 2.2b. The sensitivity of a count query is* 1 *since the addition/removal of a tuple from the data set can change the final result of the*

query by maximum 1 *(tuple). Figure* **??** *shows how the Laplace distribution for the true count in Montmartre at* t_1 *looks like. Figure 2.11b shows all the perturbed counts that are going to be released.*

2.3 Data correlation

In this Section we study the most prominent types of correlation, practices for extracting correlation from continuous data, privacy risks of correlation with a special emphasis on temporal correlation.

2.3.1 Types of correlation

The most prominent types of correlation are:

- *Temporal* [Wei06]—appearing in observations (i.e., values) of the same object over time.
- *Spatial* [Leg93, Ans95]—denoted by the degree of similarity of nearby data points in space, and indicating if and how phenomena relate to the (broader) area where they take place.
- *Spatiotemporal*—a combination of the previous categories, appearing when processing time series or sequences of human activities with geolocation characteristics, e.g., [GDSB09].

Contrary to one-dimensional correlation, spatial correlation is multi-dimensional and multi-directional, and can be measured by indicators (e.g., *Moran's I* [Mor50]) that reflect the *spatial association* of the concerned data. Spatial autocorrelation has its foundations in the *First Law of Geography* stating that "everything is related to everything else, but near things are more related than distant things" [Tob70]. A positive spatial autocorrelation indicates that similar data are *clustered*, a negative that data are dispersed and are close to dissimilar ones, and when close to zero, that data are *randomly arranged* in space.

2.3.2 Extraction of correlation

A common practice for extracting correlation from continuous data with dependence, is by expressing the data as a *stochastic* or *random process*. A random process is a collection of *random variables* or *bivariate data*, indexed by some set, e.g., a series of timestamps, a Cartesian plane $\mathbb{R}^2$, an n-dimensional Euclidean space, etc. [Sko05]. The values a random variable can take are outcomes of an

unpredictable process, while bivariate data are pairs of data values with a possible association between them. Expressing data as stochastic processes allows their modeling depending on their properties, and thereafter the discovery of relevant data dependence.

Some common stochastic processes modeling techniques include:

- *Conditional probabilities* [Gut13]—probabilities of events in the presence of other events.
- *Conditional Random Fields* (CRFs) [LMP01]—undirected graphs encoding conditional probability distributions.
- *Markov processes* [RW00]—stochastic processes for which the conditional probability of their future states depends only on the present state and it is independent of its previous states (*Markov assumption*). We highlight the following two sub-categories:
  - *Markov chains* [Gag17]—sequences of possible events whose probability depends on the state attained in the previous event.
  - *Hidden Markov Models* (HMMs) [BP66]—statistical Markov models of Markov processes with unobserved states.

2.3.3 Privacy risks of correlation

Correlation appears in dependent data:

- within one data set, and
- among one data set and previous data releases, and/or other external sources [KM11, CFYD14, LCM16, ZZP17].

In the former case, data tuples and data values within a data set may be correlated, or linked in such a way that information about one person can be inferred even if the person is absent from the database. Consequently, in this category we put assumptions made on the data generation model based on randomness, like the random world model, the independent and identically distributed data (i.i.d.) model, or the independent-tuples model, which may be unrealistic for many real-world scenarios. This attack is also known as the *deFinetti's attack* [Kif09].

In the latter case, the strength of the dependence between a pair of variables can be quantified with the utilization of *correlation* [Sti89]. Correlation implies dependence but not vice versa, however, the two terms are often used as synonyms. The correlation among nearby observations, i.e., the elements in a series of data

points, are referenced as *autocorrelation* or *serial correlation* [PP18]. Depending on the evaluation technique, e.g., *Pearson's correlation coefficient* [Sti89], a correlation can be characterized as *negative*, *zero*, or *positive*. A negative value shows that the behavior of one variable is the *opposite* of that of the other, e.g., when the one increases the other decreases. Zero means that the variables are not linked and are *independent* of each other. A positive correlation indicates that the variables behave in a *similar* manner, e.g., when the one decreases the other decreases as well.

Wand et al. [WXJ+21] examined why current differential privacy methods that either increase the noise size to offset the privacy leakage caused by the correlation (model-based) or transform correlated data into independent series to another domain and process them independently (transform-based) are inapplicable for correlated data publishing. They prove that the privacy distortion, which they quantify using entropy, after filtering out the independent and identically distributed noise from the correlated data by utilizing the data correlation (correlation-distinguishability attack) is equal to that of conditional probability inference. They conclude that the problem stems from the difference of correlation between the noise that the current methods inject and the output data.

2.3.4 Privacy loss under temporal correlation

Cao et al. [CYXX17] propose a method for computing the temporal privacy loss (TPL) of a differential privacy mechanism in the presence of temporal correlation and background knowledge. The goal of their technique is to guarantee privacy protection and to bound the privacy loss at every timestamp under the assumption of independent data releases. It calculates the temporal privacy loss as the sum of the backward and forward privacy loss minus the default privacy loss ε of the mechanism (because it is counted twice in the aforementioned entities). This calculation is done for each individual that is included in the original data set and the overall temporal privacy loss is equal to the maximum calculated value at every timestamp. The backward/forward privacy loss at any timestamp depends on the backward/forward privacy loss at the previous/next timestamp, the backward/forward temporal correlation, and ε.

Definition 4 (Temporal privacy loss (TPL) [CYXX18])**.** *The potential privacy loss of a privacy mechanism at a timestamp $t \in T$ due to a series of outputs $(\boldsymbol{o}_i)_{i \in T}$ and temporal correlation in its input D_t with respect to any adversary, targeting an individual with potential data items x_t (or x'_t) and having knowledge $\mathbb{D}_t$ equal to $D_t - \{x_t\}$ (or $D'_t - \{x'_t\}$), is defined as:*

$$\alpha_t = \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i \in T}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i \in T} | x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i \in T} | x'_t, \mathbb{D}_t]} \tag{2.1}$$

By analyzing Equation 2.1 we get the following:

$$
\begin{aligned}
(2.1) = & \underbrace{\sup_{x_t, x_t', (\boldsymbol{o}_i)_{i\in[\min(T),t]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|x_t', \mathbb{D}_t]}}_{\text{Backward privacy loss } (\alpha_t^B)} \\
& + \underbrace{\sup_{x_t, x_t', (\boldsymbol{o}_i)_{i\in[t,\max(T)]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|x_t', \mathbb{D}_t]}}_{\text{Forward privacy loss } (\alpha_t^F)} \\
& - \underbrace{\sup_{x_t, x_t', \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t, \mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x_t', \mathbb{D}_t]}}_{\text{Present privacy loss } (\varepsilon_t)}
\end{aligned} \tag{2.2}
$$

Definition 5 (Backward privacy loss (BPL) [CYXX18])**.** *The potential privacy loss of a privacy mechanism at a timestamp $t \in T$ due to outputs $(\boldsymbol{o}_i)_{i\in[\min(T),t]}$ and temporal correlation in its input D_t with respect to any adversary, targeting an individual with potential data items x_t (or x_t') and having knowledge $\mathbb{D}_t$ equal to $D_t - \{x_t\}$ (or $D_t' - \{x_t'\}$), is called backward privacy loss and is defined as:*

$$
\alpha_t^B = \sup_{x_t, x_t', (\boldsymbol{o}_i)_{i\in[\min(T),t]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|x_t', \mathbb{D}_t]} \tag{2.3}
$$

From differential privacy we have the assumption that $(\boldsymbol{o}_i)_{i\in[\min(T),t]}$ are independent events. Therefore, according to the Bayesian theorem, we can write Equation 2.3 as:

$$
\begin{aligned}
(2.3) &= \sup_{x_t, x_t', (\boldsymbol{o}_i)_{i\in[\min(T),t]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t, \mathbb{D}_t] \Pr[\boldsymbol{o}_t|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t', \mathbb{D}_t] \Pr[\boldsymbol{o}_t|x_t', \mathbb{D}_t]} \\
&= \sup_{x_t, x_t', (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t', \mathbb{D}_t]} \\
&\quad + \sup_{x_t, x_t', \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t, \mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x_t', \mathbb{D}_t]}
\end{aligned} \tag{2.4}
$$

Applying the law of total probability to the first term of Equation 2.4 for all the possible data x_{t-1} (or x_{t-1}') and $\mathbb{D}_{t-1}$ we get the following:

$$(2.4) = \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{x_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t,\mathbb{D}_t,x_{t-1},\mathbb{D}_{t-1}]\Pr[x_{t-1},\mathbb{D}_{t-1}|x_t,\mathbb{D}_t]}{\sum_{x'_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x'_t,\mathbb{D}_t,x'_{t-1},\mathbb{D}_{t-1}]\Pr[x'_{t-1},\mathbb{D}_{t-1}|x'_t,\mathbb{D}_t]}$$
$$+ \sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t,\mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t,\mathbb{D}_t]} \tag{2.5}$$

Since $\mathbb{D}_t$ is equal to $D_t - \{x_t\}$ (or $D'_t - \{x'_t\}$), and thus is constant and independent of every possible x_t (or x'_t), $\forall t \in T$, Equation 2.5 can be written as:

$$\begin{aligned}(2.5) &= \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{x_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t,\mathbb{D}_t,x_{t-1},\mathbb{D}_{t-1}]\Pr[x_{t-1}|x_t,\mathbb{D}_t]\Pr[\mathbb{D}_{t-1}|x_t,\mathbb{D}_t]}{\sum_{x'_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x'_t,\mathbb{D}_t,x'_{t-1},\mathbb{D}_{t-1}]\Pr[x'_{t-1}|x'_t,\mathbb{D}_t]\Pr[\mathbb{D}_{t-1}|x'_t,\mathbb{D}_t]} \\ &\quad + \sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t,\mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t,\mathbb{D}_t]} \\ &= \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{x_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t,\mathbb{D}_t,x_{t-1},\mathbb{D}_{t-1}]\Pr[x_{t-1}|x_t]\Pr[\mathbb{D}_{t-1}|\mathbb{D}_t]}{\sum_{x'_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x'_t,\mathbb{D}_t,x'_{t-1},\mathbb{D}_{t-1}]\Pr[x'_{t-1}|x'_t]\Pr[\mathbb{D}_{t-1}|\mathbb{D}_t]} \\ &\quad + \sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t,\mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t,\mathbb{D}_t]} \\ &= \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{x_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_t,\mathbb{D}_t,x_{t-1},\mathbb{D}_{t-1}]\Pr[x_{t-1}|x_t]}{\sum_{x'_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x'_t,\mathbb{D}_t,x'_{t-1},\mathbb{D}_{t-1}]\Pr[x'_{t-1}|x'_t]} \\ &\quad + \sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t,\mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t,\mathbb{D}_t]}\end{aligned} \tag{2.6}$$

The outputs $(\boldsymbol{o}_i)_{i\in[\min(T),t]}$ and x_t (or x'_t) are conditionally independent in the presence of x_{t-1} (or x'_{t-1}), and thus Equation 2.6 can be written as:

$$(2.6) = \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{x_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x_{t-1},\mathbb{D}_{t-1}]\Pr[x_{t-1}|x_t]}{\sum_{x'_{t-1}} \underbrace{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|x'_{t-1},\mathbb{D}_{t-1}]}_{\alpha^B_{t-1}} \underbrace{\Pr[x'_{t-1}|x'_t]}_{P^B_{t-1}}}$$
$$+ \underbrace{\sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t,\mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t,\mathbb{D}_t]}}_{\varepsilon_t} \tag{2.7}$$

Definition 6 (Forward privacy loss (FPL) [CYXX18])**.** *The potential privacy loss of a privacy mechanism at a timestamp $t \in T$ due to outputs $(\boldsymbol{o}_i)_{i\in[t,\max(T)]}$ and temporal correlation in its input D_t with respect to any adversary, targeting an*

individual with potential data item x_t (or x'_t) and having knowledge $\mathbb{D}_t$ equal to $D_t - \{x_t\}$ (or $D'_t - \{x'_t\}$), is called forward privacy loss and is defined as:

$$\alpha_t^F = \sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[t,\max(T)]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|x_t, \mathbb{D}_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|x'_t, \mathbb{D}_t]} \tag{2.8}$$

Similar to the way that we concluded to Equation 2.7 from Equation 2.3 we can write Equation 2.8 as follows:

$$\begin{aligned}(2.8) = &\sup_{x_t, x'_t, (\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}} \ln \frac{\sum_{x_{t+1}} \Pr[(\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}|x_{t+1}, \mathbb{D}_{t+1}] \Pr[x_{t+1}|x_t]}{\sum_{x'_{t+1}} \underbrace{\Pr[(\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}|x'_{t+1}, \mathbb{D}_{t+1}]}_{\alpha^F_{t+1}} \underbrace{\Pr[x'_{t+1}|x'_t]}_{P^F_{t+1}}} \\ &+ \underbrace{\sup_{x_t, x'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|x_t, \mathbb{D}_t]}{\Pr[\boldsymbol{o}_t|x'_t, \mathbb{D}_t]}}_{\varepsilon_t}\end{aligned} \tag{2.9}$$

Equations 2.2, 2.7, and 2.9 apply to the global publishing schema. In the local schema, D (or D') is a single data item and is the same with x (or x'), i.e., the possible data item of an individual user. Therefore, we calculate the extra privacy loss under temporal correlation, due to an adversary that targets a user at a timestamp t, based on the assumption that their possible data are D_t or D'_t. More specifically, the calculation of TPL (Equation 2.2) becomes:

$$\begin{aligned}&\underbrace{\sup_{D_t, D'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|D_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t]}|D'_t]}}_{\text{Backward privacy loss } (\alpha^B_t)} \\ &+ \underbrace{\sup_{D_t, D'_t, (\boldsymbol{o}_i)_{i\in[t,\max(T)]}} \ln \frac{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|D_t]}{\Pr[(\boldsymbol{o}_i)_{i\in[t,\max(T)]}|D'_t]}}_{\text{Forward privacy loss } (\alpha^F_t)} \\ &- \underbrace{\sup_{D_t, D'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|D_t]}{\Pr[\boldsymbol{o}_t|D'_t]}}_{\text{Present privacy loss } (\varepsilon_t)}\end{aligned} \tag{2.10}$$

The calculation of BPL (Equation 2.7) becomes:

$$\sup_{D_t, D'_t, (\boldsymbol{o}_i)_{i\in[\min(T),t-1]}} \ln \frac{\sum_{D_{t-1}} \Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|D_{t-1},] \Pr[D_{t-1}|D_t]}{\sum_{D'_{t-1}} \underbrace{\Pr[(\boldsymbol{o}_i)_{i\in[\min(T),t-1]}|D'_{t-1},]}_{\alpha^B_{t-1}} \underbrace{\Pr[D'_{t-1}|D'_t]}_{P^B_{t-1}}}$$

$$+ \underbrace{\sup_{D_t, D'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|D_t]}{\Pr[\boldsymbol{o}_t|D'_t]}}_{\varepsilon_t} \tag{2.11}$$

The calculation of FPL (Equation 2.9) becomes:

$$\sup_{D_t, D'_t, (\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}} \ln \frac{\sum_{D_{t+1}} \Pr[(\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}|D_{t+1}] \Pr[D_{t+1}|D_t]}{\sum_{D'_{t+1}} \underbrace{\Pr[(\boldsymbol{o}_i)_{i\in[t+1,\max(T)]}|D'_{t+1}]}_{\alpha^F_{t+1}} \underbrace{\Pr[D'_{t+1}|D'_t]}_{P^F_{t+1}}}$$

$$+ \underbrace{\sup_{D_t, D'_t, \boldsymbol{o}_t} \ln \frac{\Pr[\boldsymbol{o}_t|D_t]}{\Pr[\boldsymbol{o}_t|D'_t]}}_{\varepsilon_t} \tag{2.12}$$

The authors propose solutions to bound the temporal privacy loss, under the presence of weak to moderate correlation, in both finite and infinite data publishing scenarios. In the latter case, they try to find a value for ε for which the backward and forward privacy loss are equal. In the former, they similarly try to balance the backward and forward privacy loss while they allocate more ε at the first and last timestamps, since they have higher impact on the privacy loss of the next and previous ones. This way they achieve an overall constant temporal privacy loss throughout the time series.

According to the technique's intuition, stronger correlation result in higher privacy loss. However, the loss is less when the dimension of the transition matrix, which is extracted according to the modeling of the correlation (in this work they use Markov chains), is greater due to the fact that larger transition matrices tend to be uniform, resulting in weaker data dependence. The authors investigate briefly all of the possible privacy levels; however, the solutions that they propose are applied only on the event-level. Last but not least, the technique requires the calculation of the temporal privacy loss for every individual within the data set that might prove computationally inefficient in real-time scenarios.

Chapter 3

Related work

In this chapter, we survey works that deal with privacy under continuous data publishing covering diverse use cases. We present 48 published articles spanning 16 years of research from 2006 to 2021, with 2015 being the median, based on two levels of categorization (Figure 3.1).

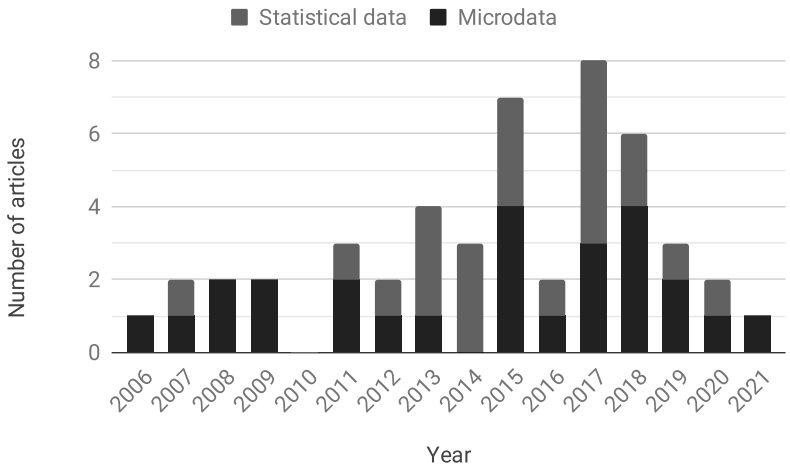
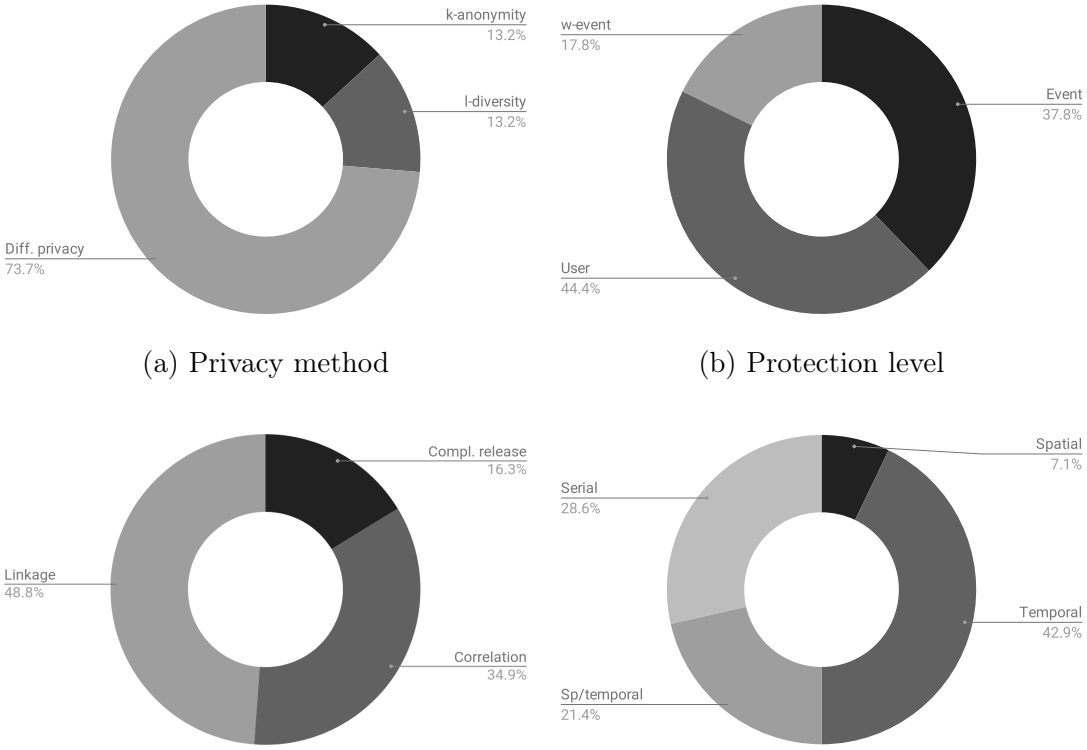


Figure 3.1: Number of reviewed published articles on continuous data publishing of microdata and statistical data per year.

First, we group works with respect to whether they deal with microdata or statistical data (see Section 2.1.1 for the definitions) as input. The works are equally

This chapter appeared in the special feature on Geospatial Privacy and Security of the 19th journal of Spatial Information Science [KTK19].

divided between the two data categories, while 55% of them propose location-specific techniques. Then, we further group them into two subcategories, whether they are designed for the finite or infinite (see Section. 2.1.2) observation setting. 59% of the reviewed literature deals with finite data observation, 57% implements the streaming publishing mode, while 77% applies the global publishing scheme. Finally, we identify the privacy-related aspects of each work in terms of the method and protection level that they apply, as well as the privacy attacks that they are considering with emphasis on the underlying data correlation (see Figure 3.2 for the detailed cumulative statistics).



(a) Privacy method

(b) Protection level

(c) Privacy attack

(d) Data correlation

Figure 3.2: The privacy-related aspects of the reviewed literature in terms of (a) the privacy method utilized, (b) the protection level provided, (c) the privacy attack considered, and (d) data correlation therein.

Our work, which we present subsequently in Section 4, focuses primarily on microdata for its use case. However, it is possible to deal with statistical data in specific scenarios. For simplicity, we limit the conversation in microdata and plan to investigate more diverse settings in our future work.

3.1 Microdata

Table 3.1 summarizes the literature for the Microdata category. Each reviewed work is abstractly described in this table, by its category (finite or infinite), its publishing mode (batch or streaming) and scheme(global or local), the level of privacy achieved (user, event, w-event), the attacks addressed, the privacy operation applied, and the base method it is built upon. We observe that privacy-preserving algorithms for microdata rely mostly on k-anonymity or derivatives of it. Ganta et al. [GKS08] revealed that k-anonymity methods are vulnerable to complementary release attacks (or *composition attacks* in the original publication). Consequently, the research community proposed solutions based on k-anonymity, focusing on different threats linked to continuous publication, as we review later on. However, notice that only a couple [LBS$^+$16, ST15] of the following works assume that data sets are privacy-protected *independently* of one another, meaning that the publisher is oblivious of the rest of the publications. On the other side, algorithms based on differential privacy are not concerned with so specific attacks as, by definition, differential privacy considers that the adversary may possess any kind of background knowledge. Moreover, more recent works consider also data dependencies to account for the extra privacy loss entailed by them.

We begin the discussion with the works designed for microdata as finite observations (Section 3.1.1), to continue with the infinite observations setting (Section 3.1.2).

3.1.1 Finite observation

Wang and Fung [WF06] address the problem of anonymously releasing different projections (i.e., subsets of the attributes) of the same data set in subsequent timestamps. More precisely, the authors want to protect individual information that could be revealed from joining various releases of the same data set. To do so, instead of locating the quasi-identifiers in a single release, the authors suggest that the identifiers may span the current and all previous releases of the (projections of the) data set. Then, the proposed method uses the join of the different releases on the common identifying attributes. The goal is to generalize the identifying attributes of the current release, given that previous releases are immutable. The generalization is performed in a top down manner, meaning that the attributes are initially over-generalized, and step by step are specialized until they reach the point when predefined quality and privacy requirements are met. The privacy requirement is the so-called *(X, Y)-privacy* for a threshold k, meaning that the identifying attributes in X are linked with at most k sensitive values in Y, in the join of the previously released and current data sets. The quality requirement can

Microdata							
Article	**Data**			**Protection**			
	Category	**Publishing**		**Level**	**Attack**	**Operation**	**Method**
		Mode	**Scheme**				
(***k***, ***δ***)-*anonymity* [ABN+08]	finite (sequential)	batch	global	user	complementary release	generalization, randomization	k-anonymity
Li et al. [LBS+16]	finite	batch	global	user	compl. release (unknown releases)	generalization, randomization	l-diversity
Erdogdu and Fawaz [EF15]	finite	batch/ streaming	local	user	correlation (temporal)	randomization	-
Jiang et al. [JSB+13]	finite (sequential)	batch	global	event	linkage	perturbation (Laplace)	differential privacy
Chen et al. [CFD11]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
Xiao et al. [XX15]	finite (sequential)	batch	local	user	correlation (temporal)	perturbation (multivariate Laplace)	differential privacy
Promesse [PMLB15]	finite (sequential)	batch	local	event	linkage	perturbation	-
DP-Star [GLTY18]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
FGS-Pufferfish [OQL+18]	finite (sequential)	batch	local	event	correlation (temporal)	perturbation (Laplace)	differential privacy
(X, Y)-privacy [WF06]	infinite (sequential)	batch	global	user	compl. release (join)	generalization, specialization	k-anonymity
BCF-anonymity [FWFP08]	infinite (incremental)	batch	global	user	compl. release (tuple correspondence)	generalization, specialization	k-anonymity
m-invariance [XT07]	infinite	batch	global	user	compl. release	generalization, synthetic data	l-diversity
e-equivalence [HBN11]	infinite	batch	global	user	compl. release (tuple equivalence)	generalization synthetic data	l-diversity
Shmueli and Tassa [ST15]	infinite (sequential)	batch	global	user	compl. release (unknown releases)	generalization, permutation	l-diversity
Zhou et al. [ZHP+09]	infinite	streaming	global	event	same with k-anonymity [Swe02b]	generalization, randomization	k-anonymity
MaskIt [GNG12]	infinite	streaming	local	event	correlation (temporal)	suppression	-
PLP [MZZ+17]	infinite	streaming	local	event	correlation (spatiotemporal)	suppression	-
Al-Dhubhani and Cazalas [ADC18]	infinite (sequential)	streaming	local	event	correlation (temporal)	perturbation (multivariate Laplace)	geo-indistinguishability
Ghinita et al. [GDSB09]	infinite (sequential)	streaming	local/ global	event	correlation (spatiotemporal)	generalization, perturbation	-
Ye et al. [YLX+17]	infinite (sequential)	streaming	global	event	linkage	generalization	l-diversity
Cao et al. [CYXX17] [CYXX18]	infinite	streaming	global	user/ (w-)event	correlation (temporal)	perturbation (Laplace)	differential privacy
ON-OFF privacy [NYER19] [YNER19] [YNR20] [YNER21]	infinite (sequential)	streaming	local	event	correlation (serial)	randomization	-

Table 3.1: Summary table of reviewed privacy-preserving algorithms for continuous microdata publishing.

be tuned into the framework. Namely, the authors propose three alternatives: the reduction of the class entropy [Qui14, Sha01], the notion of distortion, and the discernibility [BA05]. The anonymization algorithm for releasing a data set in the existence of a previously released data set takes into account the scalability and performance problems that a join among those two may entail. Still, when many previous releases exist, the complexity would remain high.

Fung et al. [FWFP08] introduce the problem of privately releasing continuous incremental data sets. As a reminder, the invariant of this kind of releases is that at every timestamp t_i, the records previously released at t_j $(j < i)$ are released again together with a set of new records. The authors first focus in two consecutive releases and describe three classes of possible attacks, which fall under the general category of complementary attacks. They name these attacks *correspondence attacks* because they rely on the principle that all tuples from an original data set D_1, from timestamp t_1, correspond to a tuple in the data set D_2, from timestamp t_2. Naturally, the opposite does not hold, as tuples added at t_2 do not exist in D_1. Assuming that the attacker knows the quasi-identifiers and the timestamp of the record of a person, they define the *backward*, *cross*, and *forward* (*BCF*) attacks. They show that combining two individually k-anonymized subsequent releases using one of the aforementioned attacks can lead to 'cracking' some of the records in the set of k candidate tuples rendering the privacy level lower than k. Except for the detection of cases of compromising BCF anonymity between two releases, the authors also provide an anonymization algorithm for a release $\boldsymbol{o}_2$ in the presence of a private release $\boldsymbol{o}_1$. The algorithm starts from the most possible generalized state for the quasi-identifiers of the records in D_2. Step by step, it checks which combinations of specializations on the attributes do not violate the BCF anonymity and outputs the most possible specialized version of the data set. The authors discuss how the framework extends to multiple releases and to different kinds of privacy methods (other than k-anonymity). It is worth noting that to maintain a certain quality for a release, it is essential that the delta among subsequent releases is large enough; otherwise the needed generalization level may destroy the utility of the data set.

Abul et al. [ABN+08] defined *(k, δ)-anonymity* for enabling high-quality moving-objects data sets publishing. The authors claim that the classical k-anonymity framework cannot be directly applied to such kind of data from a data-centric perspective. The traditional distortion techniques in k-anonymity, i.e., generalization or suppression, yield great loss of information. On the one hand, suppression diminishes the size of the database. On the other hand, generalization demands the existence of quasi-identifiers, the values of which are going to be generalized. In trajectories, however, all points can be equally considered as quasi-identifiers. Obviously, a generalization of all the trajectories

points would yield great levels of distortion. For this reason, a new, spatial-based distortion method is proposed. After clustering the trajectories in groups of at least k elements, each trajectory is translated into a new one, in a vicinity of a predefined threshold δ. Of course, the newly generated trajectories should still form a k-anonymous set. The authors validate their theory by experimentally showing that the resulting distance of count queries executed over a data set and its (k, δ) version, remains low. However, a comparative evaluation to existing clustering techniques, e.g., k-means would have been interesting, to better support the contributions on this part of the solution as well.

Erdogdu and Fawaz [EF15] consider the scenario where privacy-conscious individuals separate the data that they generate into sensitive, and non-sensitive. The individuals keep the former unreleased, and publish samples of the latter to a service provider. Privacy mapping, implemented as a stochastic process, distorts the non-sensitive data samples locally, and a separable distortion metric (e.g., Hamming distance) calculates the discrepancy of the distorted data from the original. The goal of the privacy mapping is to find a balance between the distortion and privacy metric, i.e., achieve maximum released data utility, while offering sufficient privacy guarantees. The authors assume that there is a data dependence (modeled with an HMM) between the two data sets, and thus the release of the distorted data set can reveal information about the sensitive one. They investigate both a simple attack setting, and a complex one. In the simple attack, the adversary can make static assumptions, based only on the so far made observations that cannot be later altered. In the complex attack, past, and future data releases affect dynamically the assumptions that an adversarial entity makes. In both cases, the framework quantifies the information leakage at any time point using a privacy metric that measures the improvement of the adversarial inference of the sensitive data set, which the individual kept secret, after observing the data released at that particular point. Throughout the process, the authors consider both the batch, and the streaming processing schemes. However, the assumption that individuals are privacy-conscious can drastically limit the applicability of the framework. Furthermore, the metrics that the framework utilizes for the evaluation of the privacy guarantees that it provides are not intuitive.

Xiao et al. [XT07] consider the case when a data set is (re)published in different timestamps in an update (insert/delete tuple) manner. More precisely, they address data anonymization in continuous publishing by implementing *m-invariance*. In a simple k-anonymity (or l-diverse) scenario the privacy of an individual existing in two updates can be compromised by the intersection of the set of sensitive values. In contrast, an individual who exists in a series of m-invariant releases is always associated with the same set of m different sensitive values. To enable the publishing of m-invariant data sets, artificial tuples (*counterfeits*) may be added

in a release. To minimize the noise added to the data sets, the authors provide an algorithm with two extra desiderata: limit the counterfeits, and minimize the quasi-identifiers' generalization level. Still, the choice of adding tuples with specific sensitive values disturbs the value distribution with a direct effect on any relevant statistics analysis.

In the same update setting (insert/delete tuple), He et al. [HBN11] introduce another kind of attack, namely the *equivalence* attack, not taken into account by the aforementioned m-invariance technique. The equivalence attack allows for sets of individuals to be considered equivalent as far as the sensitive attribute is concerned, in different timestamps. In this way, all the members of the equivalence class will be harmed, if the sensitive value is learned even for only one member. For a number of releases to be private, they have to be both m-invariant and e-equivalent ($e < m$). The authors propose an algorithm incorporating m-invariance, based on the graph optimization *min cut* problem, for publishing e-equivalent data sets. The proposed method can achieve better levels of privacy, in comparable times and quality as m-invariance.

Shmueli and Tassa [ST15] identified the computational inefficiency of anonymously releasing a data set, taking into account previous ones, in scenarios of continuous data publishing. The released data sets contain subsets of attributes of an original data set, while the authors propose an extension for attribute addition. Their algorithm can compute l-diverse anonymized releases (over different subsets of attributes) in parallel by generating $l - 1$ so-called *fake* worlds. A fake world is generated from the base data set by randomly permutating non-identifier and sensitive values among the tuples, in such a way that minimal information loss (quality desideratum) is incurred. This is partially accomplished by verifying that the permutation is done among quasi-identifiers that are similar. Then, the algorithm creates buckets of tuples with at least l different sensitive values, in which the quasi-identifiers will then be generalized in order to achieve l-diversity (privacy protection desideratum). The generalization step is also conducted in an information-loss efficient way. All different releases will be l-diverse because they are created assuming the same possible worlds, with which they are consistent. Tuples/attributes deletion is briefly discussed and left as an open question. The article is contrasted with a previous work [STW$^+$12] of the same authors, claiming that the new approach considers a stronger adversary (the adversary knows all individuals with their quasi-identifiers in the data set, and not only one), and that the computation is much more efficient, as it does not have an exponential complexity with respect to the number of previous publications.

Li et al. [LBS$^+$16] identified a common characteristic in most of the privacy techniques: when anonymizing a data set all previous releases are known to the data publisher. However, it is probable that the releases are independent from each

other, and that the data publisher is unaware of these releases when anonymizing the data set. In such a setting, the previous techniques would suffer from composition attacks. The authors define this kind of adversary and propose a hybrid model for data anonymization. More precisely, the publisher/adversary knows that an individual exists in two different anonymized versions of the same data set, he has a hold of the anonymized versions, but the anonymization is done independently (i.e., without considering the previously anonymized data sets) for each data set. The key idea in fighting a composition attack is to enforce the probability that the matches among tuples from two data sets are random, linking different rather than the same individual. To do so, the proposed privacy protection method exploits three preprocessing steps before applying a traditional k-anonymity or l-diversity algorithm. First, the data set is sampled so as to blur the knowledge of the existence of individuals. Then, especially in small data sets, quasi-identifiers are distorted by noise addition before the classical generalization step. The noise is taken from a normal distribution with the mean and standard deviation values calculated on the corresponding quasi-identifier values. In the case of sparse data, the sensitive values are generalized along with the quasi-identifiers. The danger of composition attacks is less prominent when using this method on top of k-anonymity rather than without, while having comparable quality results. The authors also provide a comparison to data set release using ε-differential privacy, demonstrating that their techniques are superior with respect to quality because in the opponent algorithm the noise is added up for each of the sensitive attribute to be protected. Even though the authors use in the experiments two different values for ε, a better experiment would have been to compare the quality/privacy ratio between the two methods. This is a good attempt to independently anonymize multiple times the same data set; nevertheless, the scenario is restricted to releases over the same database schema, using the same perturbation, and generalization functions.

Jiang et al. [JSB+13] focus on ship trajectories with known starting and terminal points. More specifically, they study different noise addition mechanisms for publishing trajectories with differential privacy guarantees. These mechanisms include adding global noise to the trajectory, and local noise to either each location point or the coordinates of each point of the trajectory. The first two mechanisms sample noisy radius from an exponential distribution, while the latter adds noise drawn from a Laplace distribution to each coordinate of every location. By comparing these different techniques, they conclude that the latter offers better privacy guarantee and smaller error bound. Nonetheless, the resulting trajectory is noticeably distorted due to the addition of Laplace noise to the original coordinates. To tackle this issue, they design the *Sampling Distance and Direction* (SDD) mechanism. This mechanism allows the publishing of optimal next possible

trajectory point by sampling, from the probability distribution of the exponential mechanism, a suitable distance and direction at the current position, while taking into account the ship's maximum speed constraint. Due to the fact that SDD utilizes the exponential mechanism, it outperforms the other three mechanisms, and maintains a good utility-privacy balance.

Chen et al. [CFD11] propose a non-interactive data-dependent privacy-preserving algorithm to generate a differentially private release of trajectory data. The algorithm relies on a noisy prefix tree, i.e., an ordered search tree data structure used to store an associative array. Each node represents a location, from a set of possible locations that any user can be present at, of a trajectory and contains a perturbed count, which represents the number of individuals at the current location, with noise drawn from a Laplace distribution. The privacy budget is equally allocated to each level of the tree representing a timestamp. At each level, and for every node, the algorithm seeks for the children nodes with non-zero number of trajectories (non-empty nodes) to continue expanding them. An empty node has a noisy count lower than a threshold that is dependent on the available privacy budget and the height of the tree. All children nodes associate with disjoint data subsets, and thus the algorithm can utilize for every node all of the available budget at every tree level, according to the parallel composition theorem of differential privacy. To generate the anonymized database, it is necessary to traverse the prefix tree once in post-order, paying attention to terminating (empty) nodes. During this process, taking into account some consistency constraints helps to avoid erroneous trajectories due to the noise injection. Namely, each node of a path should have a count that is greater than or equal to the counts of its children, and each node of a path should have a count that is greater than the sum of the counts of all of its children. Increasing the privacy budget results in less average relative error because less noise is added at each level, and thus improves quality. By increasing the height of the tree, the relative error initially decreases as more information is retained from the database. However, after a certain threshold, the increase of height can result in less available privacy budget at each level, and thus more relative error due to the increased perturbation.

Xiao et al. [XX15] propose another privacy definition based on differential privacy that accounts for temporal correlations in geo-tagged data. Location transitions between two consecutive timestamps are determined by temporal correlations modeled through a Markov chain. A *δ-location* set includes all the probable locations a user might appear at, excluding locations of low probability. Therefore, the true location is hidden in the resulting set, in which any pair of locations are indistinguishable. The lower the value of δ, the more locations are included and hence, the higher the level of privacy that is achieved. The authors use the *Pla-*

nar Isotropic Mechanism (PIM) as perturbation mechanism, which they designed upon their proof that l_1-norm sensitivity fails to capture the exact sensitivity in a multidimensional space. For this reason, PIM utilizes instead *sensitivity hull*, an independent notion of the context of location privacy. In [XXZC17], the authors demonstrate the functionality of their system *LocLok*, which implements the concept of δ-location.

Primault et al. [PMLB15] proposed *Promesse*, an algorithm that builds on time distortion instead of location distortion when releasing trajectories. Promesse takes as input an individual's mobility trace comprising of a data set of pairs of geolocations and timestamps, and a parameter ε. The latter indicates the desired distance between the location points that will be publicly released. Initially, Promesse extracts regularly spaced locations, and interpolates each one of the locations at a distance depending on the previous location and the value of ε. Then, it removes the first and last locations of the mobility trace, and assigns uniformly distributed timestamps to the remaining locations of the trajectory. Hence, the resulting trace has a smooth speed, and therefore places where the individual stayed longer, e.g., home, work, etc., are indistinguishable. The algorithm needs to know the starting and ending point of the trajectory; thus, it can only apply to offline scenarios. Furthermore, it works better with fine grained data sets because in this way it can achieve optimal geolocation and timestamp pairing. Moreover, the definition of ε cannot provide versatile privacy protection since it is data dependent.

Gursoy et al. [GLTY18] designed *DP-Star*, a differential privacy framework that publishes synthetic trajectories featuring similar statistics compared to the original ones. By utilizing the *Minimum Description Length* (MDL) principle [Grü07], DP-Star eliminates redundant data points in the original trajectories, and generates trajectories containing only representative points. In this way, it is necessary to allocate the available privacy budget to far less data points, striking a balance between preciseness and conciseness. Moreover, the algorithm constructs a density-aware grid, with granularity that adapts to the geographical density of the trajectory points of the data set and preserves the spatial density despite any necessary perturbation. Then, DP-Star preserves the dependence between the trajectories' start and end points by extracting (through a first-order Markov mobility model) the trip distribution, and the intra-trajectory mobility. Finally, a Median Length Estimation (MLE) mechanism approximates the trajectories' lengths, and the framework generates privacy and utility preserving synthetic trajectories. Every phase of the process consumes some predefined privacy budget, keeping the respective products of each phase private and eligible for publishing. The authors compare their design with that of [CAC12] and [HCM+15] by running several tests, and ascertain that it outperforms them in terms of data utility. However, due to DP-Star's privacy budget distribution to its different phases, for small values of ε

the framework's privacy performance is inferior to that of its competitors.

Ou et al. [OQL+18] designed *FGS-Pufferfish* for publishing temporally correlated trajectory data while protecting temporal correlation. FGS-Pufferfish transforms a user's daily trajectories into a set of sine and cosine waves of different frequencies along with the corresponding Fourier coefficients. Then, it adds Laplace noise to the Fourier coefficients' geometric sum. The authors obtain the optimal noisy Fourier coefficients by solving the constrained optimization problem via the Lagrange Multiplier method depending on the available privacy budget. They evaluate both the location data utility and the temporal correlation utility. The experimental evaluation shows that FGS-Pufferfish outperforms CTS-DP [WX17] in terms of the trade-off between privacy and location utility.

3.1.2 Infinite observation

Zhou et al. [ZHP+09] introduce the problem of infinite private data publishing, and propose a randomized solution based on k-anonymity. More precisely, they continuously publish equivalence classes of size greater than or equal to k containing generalized tuples from distinct persons (or identifiers in general). To create the equivalence classes they set several desiderata. Except for the size of a class, which should be greater than or equal to k, the information loss occurred by the generalization should be minimal, whereas the delay in forming and publishing the class should be kept low as well. To achieve these requirements, they built a randomized model using the popular structure of R-trees, extended to accommodate data density distribution information. In this way, they achieve a better quality/publishing delay ratio for the released private data. On the one hand, the formed classes contain data items that are close to each other (in dense areas), while on the other hand, classes with tuples of sparse areas are released as soon as possible so that the delay will remain low.

Gotz et al. [GNG12] developed *MaskIt*, a system that interfaces the sensors of a personal device, identifies various sets of contexts, and releases a stream of privacy-preserving contexts to untrusted applications installed on the device. A context represents the circumstances that form the setting for an event, e.g., 'at the office', 'running', etc. The individuals have to define the sensitive contexts that they wish to be protected, and the desired level of privacy. The system models the individuals' various contexts, and transitions between them. It captures temporal correlations, and models individuals' movement in the space using Markov chains while taking into account historical observations. After the initialization, MaskIt filters a stream of individual's contexts by checking for each context whether it is safe to release it or it is necessary to suppress it. The authors define *δ-privacy* as the privacy model of MaskIt. More specifically, a system preserves δ-privacy if the difference between the posterior and prior knowledge of an adversary after

observing an output at any possible timestamp is bounded by δ. After filtering all the elements of an input stream, MaskIt releases an output sequence for a single day. The system can repeat the process to publish longer context streams. The expected number of released contexts quantifies the utility of the system.

Ma et al. [MZZ+17] propose *PLP* (Protecting Location Privacy), a crowdsensing scheme that protects location privacy against adversaries that can extract spatiotemporal correlations from crowdsensing data. PLP filters an individual's context (location, sensing data) stream while it takes into consideration long-range dependencies among locations and reported sensing data, which are modeled by CRFs. It suppresses sensing data at all sensitive locations while data at non-sensitive locations are reported with a certain probability defined by observing the corresponding CRF model. On the one hand, the scheme estimates the privacy of the reported data by the difference δ between the probability that an individual would be at a specific location given the supplementary information versus the same probability without the extra information. On the other hand, it quantifies the utility by measuring the total amount of reported data (more is better). An estimation algorithm searches for the optimal strategy that maximizes utility while preserving a predefined privacy threshold.

Al-Dhubhani and Cazalas [ADC18] propose an adaptive privacy-preserving technique based on geo-indistinguishability, which adjusts the amount of noise required to obfuscate an individual's location based on its correlation level with the previously published locations. Before adding noise, an evaluation of the adversary's ability to estimate an individual's position takes place. This process utilizes a regression algorithm for a certain prediction window that exploits previous location releases. More concretely, in areas with locations presenting strong correlations, an adversary can predict the current location with low estimation error. Consequently, it is necessary to add more noise to the locations prior to their release. Adapting the amount of injected noise depending on the data correlation level might lead to a better performance, in terms of both privacy and utility, in the short term. However, alternating the amount of injected noise at each timestamp, without ensuring the preservation of the features (including correlations) present in the original data, might lead to arbitrary utility loss.

Ghinita et al. [GDSB09] tackle attacks to location privacy that arise from the linkage of maximum velocity with cloaked regions when using an LBS. The authors propose methods that can prevent the disclosure of the exact location coordinates of an individual, and bound the association probability of an individual to a sensitive location-related feature. The first method is based on temporal cloaking and utilizes deferral, and postdating. Deferral delays the disclosure of a cloaked region that is impossible for an individual to have reached based on the latest region that she published and her known maximum speed. Postdating

reports the nearest previous cloaked region that will allow the LBS to return relevant results with high probability, since the two regions are close. The second method implements spatial cloaking. First, it creates cloaked regions by taking into account all of the user-specified sensitive features that are relevant to the current location (filtering of features). Then, it enlarges the area of the region to satisfy the privacy requirements (cloaking). Finally, it defers the publishing of the region until it includes the current timestamp (safety enforcement) similar to temporal cloaking. The system measures the quality of service of both methods in terms of the cloaked region size, time and space error, and failure ratio. The cloaked region size is important because larger regions may decrease the utility of the information that the LBS might return. The time and space error is possible due to delayed location reporting and region cloaking. Failure ratio corresponds to the percentage of dropped queries in cases where it is impossible to satisfy the privacy requirements. Although both methods experimentally prove to offer adequate quality of service, the privacy requirements and metrics that the authors consider do not offer substantial privacy guarantees for commercial application.

Ye et al. [YLX+17] present an l-diversity method for producing a cloaked area, based on the local road network, for protecting trajectories. A trusted entity divides the spatial region of interest based on the density of the road network, using quadtree structures, until every subregion contains at least l road segments. Then, it creates a database for each subregion by generating all the possible trajectories based on real road network information. The trusted entity uses this database, when individuals attempt to interact with an LBS by sending their current location, to predict their next locations. Thereafter, it selects the $l-1$ nearest trajectories to the individual's current location, and constructs a minimum cloaking region. The resulting cloaking area covers the l nearest trajectories and ensures a minimum area of coverage. This method addresses the limitations of k-anonymity in terms of continuous data publishing of trajectories. The required calculation of every possible trajectory, for the construction of a trajectory database for every subregion, might require an arbitrary amount of computations depending on the area's features. Nonetheless, the utilization of quadtrees can limit the overhead of the searching process.

Cao et al. [CYXX17, CYXX18] propose a method for computing the temporal privacy loss of a differential privacy mechanism in the presence of temporal correlations and background knowledge. The goal of their technique is to guarantee privacy protection and to bound the privacy loss at every time point under the assumption of independent data releases. It calculates the temporal privacy loss as the sum of the backward and forward privacy loss minus the default privacy loss ε of the mechanism (because it is counted twice in the aforementioned entities). This calculation is done for each individual that is included in the original

data set, and the overall temporal privacy loss is equal to the maximum calculated value at every time point. The backward/forward privacy loss at any time point depends on the backward/forward privacy loss at the previous/next instance, the backward/forward temporal correlations, and ε. The authors propose solutions to bound the temporal privacy loss, under the presence of weak to moderate correlations, in both finite and infinite data publishing scenarios. In the latter case, they try to find a value for ε for which the backward and forward privacy loss are equal. In the former, they similarly try to balance the backward and forward privacy loss while they allocate more ε at the first and last time points, since they have higher impact on the privacy loss of the next and previous ones. This way they achieve an overall constant temporal privacy loss throughout the time series. According to the technique's intuition, stronger correlations result in higher privacy loss. However, the loss is smaller when the dimension of the transition matrix, which is extracted according to the modeling of the correlations (here it is Markov chain), is larger due to the fact that larger transition matrices tend to be uniform, resulting in weaker data dependence. The authors investigate briefly all of the possible privacy levels; however, the solutions that they propose are suitable only for the event-level. Last but not least, the technique requires the calculation of the temporal privacy loss for every individual within the data set which might prove computationally inefficient in real-time scenarios.

Naim et al. [NYER19, YNER19, YNR20, YNER21] proposed the notion of *ON-OFF privacy* according to which, users require privacy protection only at certain timestamps over time. They investigate the privacy risk due to the correlation between a user's requests when toggling the privacy protection ON and OFF. The goal is to minimize the information throughput and always answer users' requests while protecting their requests to online services when privacy is set to ON. They model the dependence between requests using a Markov chain, which is publicly known, where each state represents an available service. Setting privacy to ON, the user obfuscates their original query by randomly sending requests to (and receiving answers from) a subset of all of the available services. Although this randomization step makes the original query indistinguishable while making sure that the users always get the information that they need, there is no clear quantification of the privacy guarantee that the scheme offers over time.

Our work is directly applicable to microdata, and thus it applies to most of the scenarios that we discussed in this section. Most microdata methods in continuous data publishing rely on k-anonymity and its derivatives, and therefore their main point of failure is the linkage and background knowledge related attacks. Since we base our privacy notion on differential privacy, we can efficiently tackle this challenge. Finally, quite a few of the reviewed article consider data dependence and particularly temporal correlation, which is inherent in continuous data publishing.

3.2 Statistical data

As in Section 3.1, we summarize the literature for the Statistical Data category in Table 3.2, which we structure identically as Table 3.1. For a reminder, each reviewed work is abstractly described in this table, by its category (finite or infinite), its publishing mode (batch or streaming) and scheme(global or local), the level of privacy achieved (user, event, w-event), the attacks addressed, the privacy operation applied, and the base method it is built upon.

As witnessed in Table 3.2, when continuously publishing statistical data, usually in the form of counts, the most widely used privacy method is differential privacy, or derivatives of it. In theory differential privacy makes no assumptions about the background knowledge available to the adversary. In practice, data dependencies (e.g., correlations) arising in the continuous publication setting are frequently (but without it being the rule) considered as attacks in the proposed algorithms.

We begin the discussion with the works designed for microdata as finite observations (Section 3.2.1), to continue with the infinite observations setting (Section 3.2.2).

3.2.1 Finite observation

Kellaris et al. [KP13] pointed out that in time series, where users might contribute to an arbitrary number of aggregates, the sensitivity of the query answering function is significantly influenced by their presence/absence in the data set. Thus, the Laplace perturbation algorithm, commonly used with differential privacy, may produce meaningless data sets. Furthermore, under such settings, the discrete Fourier transformation of the Fourier perturbation algorithm (another popular technique for data perturbation) may behave erratically, and affect the utility of the outcome of the mechanism. For this reason, the authors proposed their own method involving grouping and smoothing for one-time publishing of time series of non-overlapping counts, i.e., the aggregated data of one count does not affect any other count. Grouping includes partitioning the data set into similar clusters. The size and the similarity measure of the clusters are data dependent. Random grouping consumes less privacy budget, as there is minimum interaction with the original data. However, when using a grouping technique based on sampling, which has some privacy cost but produces better groups, the impact of the perturbation is decreased. During the smoothing phase, the average values for each cluster are calculated, and finally, Laplace noise is added to these values. In this way, the query sensitivity becomes less dependent on each individual's data, and therefore less perturbation is required.

Chen et al. [CAC12] exploit a text-processing technique, the *n-gram* model,

Statistical data							
Article	**Data**			**Protection**			
	Category	**Publishing**		**Level**	**Attack**	**Operation**	**Method**
		Mode	**Scheme**				
Kellaris et al. [KP13]	finite	batch	global	event	linkage	perturbation (Laplace)	differential privacy
Chen et al. [CAC12]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
Hua et al. [HGZ15]	finite (sequential)	batch	global	user	linkage	perturbation (exponential, Laplace)	differential privacy
Li et al. [LZZX17]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
DPT [HCM+15]	finite (sequential)	batch	global	user	correlation (spatial)	perturbation (Laplace)	differential privacy
Song et al. [SWC17]	finite	batch	global	event	correlation	perturbation (Laplace)	pufferfish
Fan et al. [FXS13]	finite (sequential)	streaming	global	user	correlation (spatiotemporal)	perturbation (Laplace)	differential privacy
FAST [FX14]	finite	streaming	global	user	linkage	perturbation (Laplace)	differential privacy
CTS-DP [WX17]	finite	streaming	global	event	correlation (serial)	perturbation (Laplace)	differential privacy
Chan et al. [CSS11]	finite/ infinite	streaming	global	event	linkage	perturbation (Laplace)	differential privacy
l-trajectory [CY15]	infinite (sequential)	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
Bolot et al. [BFM+13]	infinite	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
Kellaris et al. [KPXP14]	infinite	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
RescueDP [WZL+16]	infinite	streaming	global	w-event	correlation (serial)	perturbation (Laplace)	differential privacy
RAPPOR [EPK14]	infinite	streaming	local	user	linkage	randomization (randomized response)	differential privacy
PrivApprox [QBB+17]	infinite	streaming	global	event	linkage	randomization (randomized response)	differential privacy
Li et al. [LSP+07]	infinite	streaming	global	event	correlation (serial)	randomization	-
PeGaSus [CMHM17]	infinite	streaming	global	event	linkage	perturbation (Laplace)	differential privacy
Errounda et al. [EL18]	infinite (sequential)	streaming	local	w-event	linkage	randomization (randomized response), perturbation (Laplace)	differential privacy
DP-PSP [WSN18]	infinite (sequential)	streaming	global	w-event	linkage	perturbation (exponential, Laplace)	differential privacy
RPTR [MZL+19]	infinite (sequential)	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
Farokhi [Far20]	infinite	streaming	global	user	linkage	perturbation (Laplace)	differential privacy

Table 3.2: Summary table of reviewed privacy-preserving algorithms for continuous statistical data publishing.

i.e., a contiguous sequence of n items from a given data sample, to release sequential data without releasing the noisy statistics (counts) of all of the possible sequences. This model allows the publishing of the most common n-grams (n is, typically, less than 5) to accurately reconstruct the original data set. The privacy technique that the authors propose is suitable for count queries and frequent sequential pattern mining scenarios. In particular, one of the applications that the authors consider concerns sequential spatiotemporal data (i.e., trajectories) of individuals. They group grams based on the similarity of their n values, construct a search tree, and inject Laplace noise to each node value (count) to achieve user-level differential privacy protection. Instead of allocating the available privacy budget based on the overall maximum height of the tree, they estimate each path adaptively based on known noisy counts. The grouping process continues until the desired threshold of n is reached. Thereafter, they release variable-length n-grams with certain thresholds for the values of counts and tree heights, allowing to deal with the trade-off of shorter grams having less information than longer ones but less relative error. They use a set of consistency constraints, i.e., the sum of each node's noisy count has to be less than or equal to its parent's noisy count, and all the noisy counts of leaf nodes have to be within a predefined threshold. These constraints improve the final data utility since they result in lower values of n. On the one hand, this translates into higher counts, large enough to deal with noise injection and the inherent Markov assumption in the n-gram model. On the other hand, it enhances privacy when the universe of all grams with a lower n value is relatively small resulting in more common sequences, which, nonetheless, is rarely valid in real-life scenarios.

Hua et al. [HGZ15] use, similar to the scheme proposed in [CAC12], the n-grams modeling technique for publishing trajectories containing a small number of n-grams, thus, sharing few or even no identical prefixes. They propose a differentially private location-specific generalization algorithm (exponential mechanism), where each position in the trajectory is one record. The algorithm probabilistically partitions the locations at each timestamp with probability proportional to their Euclidean distance from each other. They replace each partition with its centroid and therefore, they offer better utility by creating groups of locations belonging to close trajectories. They optimize the algorithm for time efficiency by using classic k-means clustering. Then, the algorithm releases the new trajectories by observing the generalized location partitions, and their perturbed counts (i.e., sum of the same locations at each timestamp) with noise drawn from a Laplace distribution. The process continues until the total count of the published trajectories reaches the size of the original data set. They can limit the total number of the possible trajectories by taking into account the individual's moving speed. The authors have measured the utility of distorted spatiotemporal range queries by measuring

the Hausdorff distance from the original results and concluded that the utility deterioration is within reasonable boundaries considering the offered privacy guarantees. Similar to [CAC12], their approach works well for a small location domain. To make it applicable to realistic scenarios, it is essential to truncate the original trajectories in an effort to reduce the location domain. This results in a coarse discretization of the location area, leading to the arbitrary distortion of the spatial correlations that are present in the original data set.

Li et al. [LZZX17] focus on publishing a set of trajectories, where, contrary to [HGZ15], each one is considered as a single entry in the data set. First, using k-means clustering they partition the original locations based on their pairwise Euclidean distances. The scheme represents each location partition by their mean (centroid). A larger number of partitions, in areas where close centroids exist, results in fewer locations in each partition, and thus lower trajectory precision loss. Before adding noise, they randomly select partition centroids to generate trajectories until they reach the size of the original data set. Then, they generate Laplace noise, which they bound according to a set of constraints, and they add it to the count of locations of each point of every trajectory. Finally, they release the generalized trajectories along with the noisy count of each location point. The authors prove experimentally that they reduce considerably the trajectory merging time at the expense of utility.

He et al. present *DPT* (Differentially Private Trajectory) [HCM+15], a system that synthesizes mobility data based on raw, speed-varying trajectories of individuals, while providing ε-differential privacy protection guarantees. The system constructs a Hierarchical Reference Systems (HRS) model to capture correlations between adjacent locations by imposing a uniform grid at multiple resolutions (i.e., for different speed values) over the space, keeping a prefix tree for each resolution, and choosing the centroids as anchor points. In each reference system, anchor points have a small number of neighboring points with increasing (by a constant factor) average distance between them, and fewer children anchor points as the grid resolution becomes finer. DPT estimates transition probabilities only for the anchor points in proximity to the last observed location, and chooses the appropriate reference system for each raw point so that the consecutive points of the trajectory are either neighboring anchors or have a parent-child relationship. The system generates the transition probabilities by estimating the counts in the prefix trees. Thereafter, it chooses the appropriate prefix trees, perturbs them with noise drawn from the Laplace distribution, and adaptively prunes subtrees with low counts to improve the resulting utility. DPT implements a direction-weighted sampling postprocessing strategy for the synthetic trajectories to avoid the loss of directionality of the original trajectories due to the perturbation. Nonetheless, as with all other similar techniques, the usage of prefix trees limits the length of the

released trajectories, which results into an uneven spatial distribution.

Song et al. [SWC17] propose the *Wasserstein mechanism*, a technique that applies to any general instantiation of Pufferfish (see Section 2.2.5). It adds noise proportional to the sensitivity of a query F, which depends on the worst case distance between the distributions $P(F(X)|s_i, d)$ and $P(F(X)|s_j, d)$ for a variable X, a pair of secrets (s_i, s_j), and an evolution scenario d. The Wasserstein metric function calculates the worst case distance between those two distributions. The noise is drawn from a Laplace distribution with parameter equal to the quotient resulting from the division of the maximum Wasserstein distance of the distributions of all the pairs of secrets by the available privacy budget ε. For optimization purposes, the authors consider a more restricted setting. This setting, utilizes an evolution scenario for the data correlations representation, and Bayesian networks for the correlation modeling. The authors state that in cases where Bayesian networks are complex, the Markov chains are a more efficient alternative. A generalization of the *Markov blanket* mechanism, the *Markov quilt* mechanism, calculates data dependencies. The dependent nodes of any node consist of its parents, its children, and the other parents of its children. The present technique excels at data sets generated by monitoring applications or networks, but it is not suitable for online scenarios.

Fan et al. [FXS13] propose a real-time framework for releasing differentially private multi-dimensional traffic monitoring data. At every timestamp, the Perturbation module injects noise drawn from a Laplace distribution to the data. Then, the Estimation module post-processes the perturbed data to improve the accuracy. The authors propose a temporal, and spatial estimation algorithm. The former estimates an internal time series model for each location to improve the utility of the perturbation's outcome by performing a posterior estimation that utilizes Gaussian approximation and Kalman filtering[Kal60]. The latter reduces data sparsity by grouping neighboring locations using a spatial indexing structure based on quadtree. The Modeling/Aggregation module utilizes domain knowledge, e.g., road network and density, and has a bidirectional interaction with the other two in parallel. Although the authors propose the framework for real-time scenarios, they do not deal with infinite data processing/publication, which limits considerably its applicability.

In another work, Fan et al. designed *FAST* [FX14], an adaptive system that allows the release of real-time aggregate time series under user-level differential privacy. These were achieved by using a Sampling, a Perturbation, and a Filtering module. The Sampling module samples on an adaptive rate the aggregates to be perturbed. The Perturbation module adds noise to each sampled point according to the allocated privacy budget. The Filtering module receives the perturbed data point and the original one and generates a posterior estimate, which is finally

released. The error between the perturbed and the released (posterior estimate) point is used to adapt the sampling rate; the sampling frequency is increased when data is going through rapid changes and vice-versa. Thus, depending on the adjusted sampling rate, not every single data point is perturbed, saving in this way the available privacy budget. While the system considers the temporal correlations of the processed time series, it does not attempt to deal with the privacy threat that they might pose.

Wang and Zu [WX17] defined Correlated Time Series Differential Privacy (*CTS-DP*). The scheme guarantees that the correlation between the perturbation that is introduced by a Correlated Laplace Mechanism (CLM), and the original time series is indistinguishable (Series-Indistinguishability). CTS-DP deals with the shortcomings of independent and identically distributed (i.i.d.) noise under the presence of correlations. I.i.d. noise offers inadequate protection, because refinement methods, e.g., filtering, can remove it. Most privacy-preserving methods choose to introduce more noise in the presence of strong correlations thus, diminishing the data utility. An original and a perturbed time series satisfy Series-Indistinguishability if their normalized autocorrelation functions are the same; hence, the two time series are indistinguishable and the published time series satisfies differential privacy as well. The authors consider the fact that, in signal processing, if an i.i.d. signal passes through a filter, which consists of a combination of adders and delayers, it becomes non-i.i.d. Hence, they design CLM, which uses four Gaussian white noise series passed through a linear system, to produce a correlated Laplace noise series according to the autocorrelation function of the original time series. Although the authors prove experimentally that the implementation of CLM outperforms the current state-of-the-art methods, they do not test its robustness against any filter, which they keep as future work.

3.2.2 Infinite observation

Chan et al. [CSS11] designed continuous counting mechanisms for finite and infinite data processing and publishing, satisfying ε-differential privacy. Their main contribution lies in proposing the Binary and Hybrid mechanisms, which do not have any upper bound temporal requirements. The mechanisms rely on the release of intermediate partial sums of counts at consecutive timestamp intervals, called *p-sums*, and the injection of noise drawn from a Laplace distribution. The Binary mechanism constructs a binary tree where each node corresponds to a p-sum, and adds noise to each released p-sum proportional to its corresponding length. The Hybrid mechanism publishes counts at sparse time intervals, i.e., timestamps that are a power of 2. Both mechanisms offer event-level protection (pan-privacy) under single unannounced and continual announced intrusions by adding a certain amount of noise to every p-sum in memory. They can facilitate continual top-k

queries in recommendation systems, and multidimensional range queries. Furthermore, they are able to support applications that require a consistent output, i.e., at each timestamp the counter increases by either 0 or 1.

Cao et al. [CY15] developed a framework that achieves personalized *l-trajectory* privacy protection by dynamically adding noise at each timestamp, which exponentially fades over time. Each individual can specify, in an array of size l, the desired protection level for each location of his/her trajectory. The proposed framework is composed of three components. The Dynamic Budget Allocation component allocates portions of the privacy budget to the other two components: a fixed one to the Private Approximation, and a dynamic one to the Private Publishing component at each timestamp. The Private Approximation component estimates, under a utility goal and an approximation strategy, whether it is beneficial to publish approximate data or not. More precisely, it chooses an appropriate previous noisy data release and republishes it if it is similar to the real statistics planned to be published. The Private Publishing component takes as inputs the real statistics, and the timestamp of the approximate data, generated by the Private Approximation component, to be republished. If the timestamp of the approximate data is equal to the current timestamp, then the current data with Laplace noise are published. Otherwise, the data at the corresponding timestamp of the approximate data will be republished. The utilized approximation technique is highly suitable for streaming processing, due to the implementation of approximation that can reduce significantly the privacy budget consumption. However, the framework does not take into account privacy leakage stemming from data dependencies, which limits considerably its applicability in real life data sets.

Bolot et al. [BFM$^+$13] introduce the notion of *decayed privacy* in continual observation of aggregates (sums). The authors recognize the fact that monitoring applications focus more on recent events, and data, therefore, the value of previous data releases exponentially fades. This leads to a schema of privacy with expiration, according to which, recent events, and data are more privacy-sensitive than those preceding. Based on this, they apply decayed sum functions for answering sliding window queries of fixed window size w on data streams. Namely, window sum compute the difference of two running sums, and exponentially decayed and polynomial decayed sums estimate the sum of decayed data. For every consecutive w data points the algorithm generates binary trees where each node is perturbed with Laplace noise with scale proportional to w. Instead of maintaining a binary tree for every window, the algorithm considers the windows that span two blocks as the union of a suffix and a prefix of two consecutive trees. This way, the global sensitivity of the query function is kept low. The proposed techniques are designed for fixed window sizes, hence, when answering multiple sliding window queries with variable window sizes they have to distribute the available privacy

budget accordingly.

Based on the notion of decayed privacy [BFM+13], Kellaris et al. [KPXP14] defined w-event privacy in the setting of periodical release of statistics (counts) in infinite streams. To achieve w-event privacy, the authors propose two mechanisms (Budget Distribution, and Budget Absorption) based on sliding windows, which effectively distribute the privacy budget to sub-mechanisms (one sub-mechanism per timestamp) applied on the data of a window of the stream. Both algorithms may decide to publish a new noisy count for a specific timestamp, based on the similarity level of the current count with a previously published one. Moreover, both algorithms have the constraint that the total privacy budget consumed in a window is less than or equal to ε. The Budget Distribution algorithm distributes the privacy budget in an exponential-fading manner following the assumption that in a window most of the counts remain similar. The budget of expired timestamps becomes available for the next publications (of next windows). The Budget Absorption algorithm uniformly distributes from the beginning the budget to the window's timestamps. A publication uses not only the by-default allocated budget but also the budget of non-published timestamps. In order to not exceed the limit of ε, adequate number of subsequent timestamps are 'silenced' after a publication takes place. Even though one can argue that w-event privacy could be achieved by user-level privacy, it is nevertheless non-practical because of the rigidity of the budget allocation that would finally render the output useless.

Wang et al. [WZL+16] propose *RescueDP* for the publishing of real-time user-generated spatiotemporal data, utilizing differential privacy with w-event-level protection. RescueDP uses a Dynamic Grouping module to create clusters of regions with small statistics, i.e., areas with a small number of samples. It estimates the similarity of the data trends of these regions by utilizing the Pearson's correlation coefficient, and creates groups accordingly. The data of each group pass from a Perturbation module that injects Laplace noise to them. The grouping of the previous phase results into the increase of the sample size of each group of regions, which minimizes the error due to the noise injection. The implementation of a Kalman Filtering [Kal60] module further increases the utility of the released data. A Budget Allocation module distributes the available privacy budget to sampling points within any successive w timestamps. RescueDP saves part of the available privacy budget by approximating the non-sampled data with previously released perturbed data. During the whole process, an Adaptive Sampling module adjusts the sampling interval according to the difference in the released data statistics over the previous timestamps while taking into account the remaining privacy budget.

Erlingsson et al. [EPK14] presented *RAPPOR* (Randomized Aggregatable Privacy-Preserving Ordinal Response) as a solution for privacy-preserving collection of statistics. RAPPOR makes all the necessary data processing on the side

of the data generators by applying the method of randomized response, which guarantees local differential privacy. The product of each local privacy-preserving processing is a report that can be represented as a bit string. Each bit corresponds to a randomized response to a logical predicate on an individual's personal data, e.g., categorical properties, numerical and ordinal values, or categories that cannot be enumerated. Initially, RAPPOR hashes a sensitive value into a Bloom filter [Blo70]. It creates a binary reporting value, which keeps in its memory (*memoization*) and reuses for future reports (permanent randomized response). Memoization offers long-term longitudinal privacy protection for privacy-sensitive data values that do not change over time or that are not dependent. RAPPOR deals with tracking externalities by reporting a randomized version of the permanent randomized response (instantaneous randomized response). Although this adds an extra layer of randomization to the reported values, it might lead to an averaging attack that may allow an adversary to estimate the true value. Finally, the authors propose a decoding technique that involves grouping, least-squares solving, and regression. This way, they effectively make up for the loss of information due to the randomization of the previous steps and allow the extraction of useful information when observing the generated bit strings. They test their implementation with both simulated and real data, and show that they can extract statistics with high utility while preserving the privacy of the individuals involved. However, the fact that the privacy guarantees of their technique are valid only for stationary individuals that produce independent data on top of the relatively complex configuration, renders their proposal impractical for many real-world scenarios.

Le Quoc et al. [QBB+17] propose *PrivApprox*, a data analytics system for privacy-preserving stream processing of distributed data sets that combines sampling and randomized response. The system distributes the analysts' queries to clients via an aggregator and proxies, and employs sliding window computations over batched stream processing to handle the data stream generated by the clients. The clients transmit a randomized response, after sampling the locally available data, to the aggregator via proxies that apply (XOR-based) encryption. The combination of sampling and randomized response achieves *zero-knowledge* based privacy, i.e., proving that they know a piece of information without in fact disclosing its actual value. The aggregator collects the received responses and returns statistics to the analysts. The query model expresses the responses of numerical queries as counts within histogram buckets, whereas, for non-numeric queries it specifies each bucket by a matching rule or a regular expression. A confidence metric quantifies the results' approximation from the sampling and randomization. PrivApprox achieves low latency stream processing and enables a synchronization-free distributed architecture that requires low trust to a central entity. Since it

implements a sliding window methodology for infinitely processing series of data sets, it would be purposeful to investigate how to achieve w-event-level privacy protection.

Li et al. [LSP+07] attempt to tackle the problem of privacy preservation in numerical data streams taking into account the correlations that may appear continuously among multiple streams and within each one of them. Firstly, the authors define the utility, and privacy specifications. The utility of a perturbed data stream is the inverse of the discrepancy between the original and the perturbed measurements. The discrepancy is set as the normalized Forbenius norm, i.e., a matrix norm defined as the square root of the sum of the absolute squares of its elements. Privacy corresponds to the discrepancy between the original and the reconstructed data stream (from the perturbed one), and consists of the removed noise and the error introduced by the reconstruction. Then, correlations come into play. The system continuously monitors the data streams for trends to track correlations, and dynamically perturbs the original numerical data while maintaining the trends that are present. More specifically, the Streaming Correlated Additive Noise (SCAN) module updates the estimation of the local principal components of the original data, and proportionally distributes noise along the components. Thereafter, the Streaming Correlation Online Reconstruction (SCOR) module removes all the noise by utilizing the best linear reconstruction. SCOR is a representation of the ability of any adversarial entity to post-process the released data and attempt to reconstruct the original data set by filtering out any distortion. Overall, the present technique offers robustness against inference attacks by adapting randomization according to data trends, but fails to efficiently quantify the overall privacy guarantee.

Chen et al. [CMHM17] developed *PeGaSus*, an algorithm for event-level differentially private stream processing that supports different categories of stream queries (counts, sliding window, and event monitoring) over multiple stream resolutions. It consists of a Perturber, a Grouper, and a Smoother modules. The Perturber consumes the incoming data stream, adds noise ε_p, which is part of the available privacy budget ε to each data item, and outputs a stream of noisy data. The data-adaptive Grouper consumes the original stream and partitions the data into well-approximated regions using, also part of the available privacy budget, ε_g. Finally, a query specific Smoother combines the independent information produced by the Perturber and the Grouper, and performs post-processing by calculating the final estimates of the Perturber's values for each partition created by the Grouper at each timestamp. The combination of the Perturber and the Grouper follows the sequential composition and post-processing properties of differential privacy, thus, the resulting algorithm satisfies $(\varepsilon_p + \varepsilon_g)$-differential privacy.

Errounda et al. [EL18] proposed a algorithm for sharing w-event local dif-

ferentially private statistics over infinite streams of location data. The decision mechanism determines the similarity between the current data of every individual and the most recent release, with respect to a predefined threshold. Using the randomized response mechanism, it perturbs the result of this comparison and decides whether to perform an approximation based on the most recent release or calculate and release the current statistics after injecting to them Laplacian noise. Within the sliding window of size w, the privacy budget allocation mechanism estimates the overall privacy budget that the algorithm has allocated at any timestamp and decides how to optimally allocate the remaining budget in the future timestamps. The evaluation of the algorithm show that, according to the relevant literature on local differential privacy, the author's work achieves the the same utility as the centralized approach of differential privacy.

Wang et al. [WSN18] presented *DP-PSP*, an approach for publishing differentially private statistics over infinite streams of trajectory data. DP-PSP segments trajectories by taking into account points of interest in road networks. A start and end point (anchor) represents a segment and each data point in the trajectory data is calibrated to the nearest anchor. This segmentation facilitates a less computationally intensive statistical processing and more efficient privacy budget allocation. The authors designed a private k nearest neighbors algorithm by utilizing the exponential mechanism, which uses the Gaussian weighted Euclidean distance for utility function, to generate the connected segments for each segment. Thus, at some timestamps, they can predict accurately the upcoming statistics, and therefore save part of the available privacy budget by releasing an approximation instead of perturbing the original data. DP-PSP allocates the available privacy budget, in an exponentially decaying fashion, in a sliding window with a user-defined size w, satisfying w-event-level privacy. Statistics over the trajectory combined with Laplacian noise are released in the end of the process by DP-PSP. From the implementation, it is not clear how DP-PSP takes into consideration all of the user preferences regarding the size of w while releasing statistics of the data of all of the sample.

Ma et al. [MZL+19] implemented *RPTR*, a w-event differential privacy mechanism for protecting statistics of vehicular trajectory data in real time. RPTR adapts the rate with which it samples data according to the accuracy with which it can predict future statistics based on historical data and position transfer probability matrix and according to how much the original data change through time based on Pearson coefficient. Before releasing data statistics, the mechanism perturbs the original values with Laplacian noise the impact of which is mitigated by using Ensemble Kalman filtering. The combination of adaptive sampling and filtering can improve the accuracy when predicting the values of non-sampled data points, and thus saving more privacy budget (i.e., higher data utility) for data

points that the mechanism decides to release. The mechanism detects highly frequented map regions and, using a quad-tree, it calculate the each region's privacy weight. In their implementation, the authors assume that highly frequented regions tend to be more privacy-sensitive, and thus more noise (i.e., less privacy budget to invest) needs to be introduced before publicly releasing the users' data falling into these regions. The efficiency (both in terms of user privacy and data utility) of the mechanism depends on the number of regions that it divides the map, and therefore the challenge of its optimal division is an interesting future research topic.

Farokhi [Far20] proposed a relaxation of the user-level protection of differential privacy based on the discounted utility theory in economics. More specifically, at each timestamp, the scheme of *temporally discounted differential privacy* assigns different weights to the privacy budgets that have been invested in previous timestamps. These weights decrease the further that we observe in the past. The author implements an exponentially and a hyperbolic discounted scheme. In the former, the discount factor, which is positive and less than 1, and in the latter, the discounting coefficient, which is greater or equal to 0, allows the adjustment of temporal discounting. Increasing the discount factor offers stronger privacy protection, equivalent to that of user-level. Whereas, increasing the discount coefficient resembles the behavior of event-level differential privacy. Selecting a suitable value for the privacy budget and the discount parameter allows for bounding the overall privacy loss in an infinite observation scenario. However, the assumption that all users discount previous data releases limits the applicability of the the current scheme in real-world scenarios for statistical data.

Most of the proposed methods in this section utilize differential privacy, on which we base our work. However, few of them account for data dependence and particularly temporal correlation, which is inherent in time series. In this thesis, we generally investigate the presence of correlation in data and we propose a method that accounts for temporal correlation throughout finite time series. Last but not least, although the use case of our work focuses on microdata, it can adapt to scenarios that requite data aggregation, and thus extend its applicability.

3.3 Summary

In this chapter, we surveyed the literature around the domain of privacy-preserving continuous data publishing in microdata and statistical data. We further categorized the works in terms the span of the data observation in finite and infinite. Moreover, we summarize the methods for each data category in tabular form (with detailed attributes) aiming to offer a guide that would allow its users to choose

the proper algorithm(s) for their specific use case. Such a documentation becomes very useful nowadays, due to the abundance of continuously user-generated data sets that could be analyzed and/or published in a privacy-preserving way, and the quick progress made in this research field.

Since the domain of data privacy is vast, several surveys have already been published with different scopes. A group of surveys focuses on specific different families of privacy-preserving algorithms and techniques. For instance, Simi et al. [SNE17] provide an extensive study of works on k-anonymity and Dwork [Dwo08] focuses on differential privacy. Another group of surveys focuses on techniques that allow the execution of data mining or machine learning tasks with some privacy guarantees, e.g., Wang et al. [WLZL09], and Ji et al. [JLE14]. In a more general scope, Wang et al. [WCFY10] analyze the challenges of privacy-preserving data publishing, and offer a summary and evaluation of relevant techniques. Additional surveys look into issues around Big Data and user privacy. Indicatively, Jain et al. [JGK16], and Soria-Comas and Domingo-Ferrer [SCDF16] examine how Big Data conflict with pre-existing concepts of privacy-preserving data management, and how efficiently k-anonymity and ε-differential privacy deal with the characteristics of Big Data. Others narrow down their research to location privacy issues. To name a few, Chow and Mokbel [CM11] investigate privacy protection in continuous LBSs and trajectory data publishing, Chatzikokolakis et al. [CEP+17] review privacy issues around the usage of LBSs and relevant protection mechanisms and metrics, Primault et al. [PBMB18] summarize location privacy threats and privacy-preserving mechanisms, and Fiore et al. [FKZ+19] focus only on privacy-preserving publishing of trajectory microdata. Finally, there are some surveys on application-specific privacy challenges. For example, Zhou et al. [ZPL08] have a focus on social networks, and Christin et al. [CRKH11] give an outline of how privacy aspects are addressed in crowdsensing applications.

Chapter 4

Landmark privacy

The plethora of sensors currently embedded in personal devices and other infrastructures have paved the way for the development of numerous *crowdsensing services* (e.g., Ring [rin21], TousAntiCovid [tou21], Waze [waz21], etc.) based on the collected personal, and usually geotagged and timestamped data. User–service interactions gather personal event-like data, which are tuples of an identifying attribute of an individual and the—possibly sensitive—information with a timestamp e.g., (*'Quackmore', 'dining', 'Canal Saint-Martin',* 17:00). When the interactions are performed in a continuous manner, we obtain *time series* of events. Example 4.0.1 is an example of a user–service interaction that results in retrieving location-based information or reporting user-state at various locations.

Example 4.0.1. *Figure 4.1 shows a finite sequence of spatiotemporal data, generated by Quackmore, during an interval of* 8 *timestamps. Events in gray correspond to significant events that Bob has defined beforehand, because they are related to his home (around Élysée), his workplace (around the Louvre), and his hangout (around Canal Saint-Martin).*

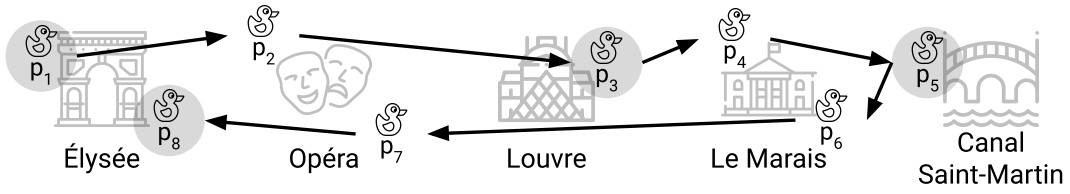


Figure 4.1: A time series with landmarks (highlighted in gray).

This chapter will appear in the proceedings of the 12th ACM conference on Data and Application Security and Privacy [KTK22].

The regulation regarding the processing of user-generated data sets [Tan16] requires the provision of privacy guarantees to the users. To accomplish this, various privacy techniques perturb the original data or their statistical output at the expense of the overall utility of the final output. Meanwhile, it is essential to provide data of high utility to the final consumers of the privacy-preserving process. A widely recognized method that introduces probabilistic randomness to the original data, while quantifying with a parameter ε ('privacy budget' [McS09]) the privacy/utility ratio, is *ε-differential privacy* [DMNS06]. Due to its *composition* property, i.e., the combination of differentially private outputs satisfies differential privacy as well, differential privacy is suitable for privacy-preserving time series publishing. *Event*, *user* [DNPR10], and *w-event* [KPXP14] comprise the possible levels of privacy protection. Event-level limits the protection to *any single event*, user-level protects *all the events* of any user, and w-event provides protection to *any sequence of w events*. In every case, privacy protection boils down to allocating to events an overall privacy budget that does not exceed ε.

In this chapter, we propose a novel configurable privacy scheme, *landmark* privacy (Section 4.1), which takes into account significant events (*landmarks*) in the time series and allocates the available privacy budget accordingly. We propose three privacy schemes that guarantee landmark privacy. To further enhance our privacy methodology, and protect the landmarks position in the time series, we propose techniques to perturb the initial landmarks set (Section 4.2).

4.1 Significant events

The privacy mechanisms for the user, w-event, and event levels that are already proposed in the literature, assume that in a time series any single event, or any sequence of events, or the entire series of events is equally privacy-significant for the users. In reality, this is an assumption that deteriorates unnecessarily the utility of the released data. The significance of an event is related to certain user-defined privacy criteria, or to its adjacent events, as well as to the entire time series. We term significant events as *landmark events* or simply *landmarks*, following relevant literature [GWO00].

Identifying landmarks in time series can be done in an automatic or manual way. For example, in spatiotemporal data, *places where an individual spent some time* denote *points of interest* (POIs) (called also stay points) [Zhe15]. Such events, and more particularly their spatial attribute values, can be less privacy-sensitive [PBMB18], e.g., parks, theaters, etc., or, if individuals frequent them, they can reveal supplementary information, e.g., residences (home addresses) [GKdPC10], places of worship (religious beliefs) [FB15], etc. POIs can be an example of how we can choose landmarks, but the idea is not limited to

these. Another example is the detection of privacy-sensitive user interactions by *contact tracing* applications. This can be practical in decease control [EK03], similar to the recent outbreak of the Coronavirus disease 2019 (COVID-19) epidemic [AMX+20]. Last but not least, landmarks in *smart grid* electricity usage patterns may not only reveal the energy consumption of a user but also information regarding activities, e.g., 'at work', 'sleeping', etc., or types of appliances already installed or recently purchased [KHLF10]. We stress out that landmark identification is an orthogonal problem to ours, and that we consider landmarks given as input to our problem.

We argue that protecting only landmark events along with any regular event is sufficient for the user privacy protection, while it improves data utility with respect to the conventional user-level privacy. Considering landmarks can prevent over-perturbing the data in the benefit of their final utility. Revisiting the scenario in Figure 4.2, if we want to protect the landmark points, we have to allocate at most a budget of ε to the landmarks, while saving some for the release of regular events. Essentially, the more budget we allocate to an event the less we protect it, but at the same time the more we maintain its utility. With landmark privacy we propose to distribute the budget by accounting only for the landmarks when we release an event of the time series, i.e., allocating $\frac{\varepsilon}{5}$ (4 landmarks +1 regular point) to each event (see Figure 4.2). This way, we still guarantee that the landmarks are adequately protected, as they receive a total budget of $\frac{4\varepsilon}{5} < \varepsilon$. At the same time, we avoid over-perturbing the regular events, as we allocate to them a higher total budget ($\frac{4\varepsilon}{5}$) than in user-level ($\frac{\varepsilon}{2}$), and thus less noise. Hence, at any timestamp we achieve an overall privacy protection bounded by ε in the event set consisting of the released event and the landmarks.

Example 4.1.1. *Continuing Example 4.0.1, Quackmore cares about protecting his landmarks (p_1, p_3, p_5, p_8) along with every release that he makes, however he is not equally interested for the other regular events in his trajectory. More technically, he cares about allocating a total budget of ε on any set of timestamps containing the landmarks and one regular event. Event-level protection is not suitable for this case, since it can only protect one event at a time. So, let us assume that we apply user-level privacy*[1]*, by distributing equal portions of ε to all the events, i.e., $\frac{\varepsilon}{8}$ to each one (see Figure 4.2). Indeed, we have protected the landmark points plus one regular event at any release as expected; we have allocated a total of $\frac{5\varepsilon}{8} < \varepsilon$ to these* 5 *events.*

However, perturbing by $\frac{\varepsilon}{8}$ each one of the regular points deteriorates the data utility unnecessarily; any budget lower than or equal to $\frac{4\varepsilon}{8}$ would be sufficient for covering the user privacy requirements. On the other hand, our proposed privacy

[1]In this scenario, in order to protect all the landmarks from timestamp 1 to 8, w must be set to 8, which makes w-event privacy equivalent to user-level.

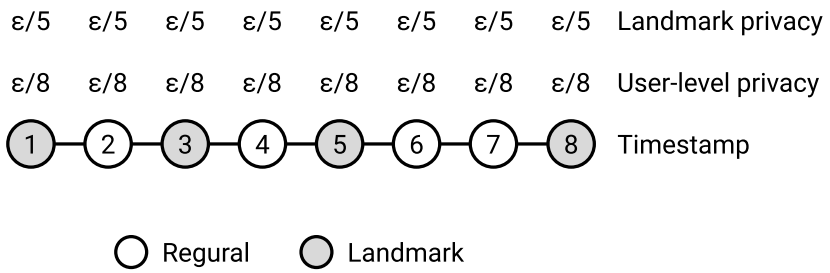


Figure 4.2: User-level and landmark ε-differential privacy protection for the time series of Figure 4.1.

model, landmark privacy, directly considers only the 5 *events of interest (*4 *landmarks* +1 *current event) in every release, thus changing the scope from all the time series to a significant subset of events. Subsequently, it allocates* $\frac{\varepsilon}{5}$ *to each one of these events. Consequently, we still achieve to protect all the significant events, while the utility of a perturbed event is higher than in the case of user-level privacy (*$\frac{\varepsilon}{5} > \frac{\varepsilon}{8}$*).*

4.1.1 Contribution

In this section, we formally define a novel privacy notion that we call *landmark privacy*. We apply this privacy notion to time series consisting of *landmarks* and regular events, and we design and implement three landmark privacy schemes. We investigate landmark privacy under temporal correlation, which is inherent in time series publishing, and discuss how landmarks can affect the propagation of temporal privacy loss.

4.1.2 Problem definition

In this section, we introduce a new privacy definition.

Setting

Our problem setting consists of three entities: (i) data generators (users), (ii) data publishers (trusted non-adversarial entities), and (iii) data consumers (possibly adversarial entities). Users generate a finite series of sensitive data over time, which are processed in batch mode in a secure and private way locally (or by a trusted curator) and are later published in order to be consumed by potentially adversarial data analysts. Data are produced as a series of events, which we call time series.

(i) **Data generators** (users) entity E_g interacts with a crowdsensing application and produces continuously privacy-sensitive data items in an arbitrary frequency during the application's usage period $T = (i)_{i \in \mathbb{N}}$. Thus, at each timestamp t, E_g generates a data set $D_i \in \mathcal{D}$ where each of its members contributes a single data item.

(ii) **Data publishers** (trusted non-adversarial) entity E_p receives the data sent by E_g in the form of a series of events in T. Following the *global* processing and publishing scheme, E_p collects at t a data set D_i and privacy-protects it by applying the respective privacy mechanism $\mathcal{M}_i$. $\mathcal{M}_i$ uses independent randomness such that it satisfies ε_i-differential privacy.

(iii) **Data consumers** (possibly adversarial) entity E_c receives the result $\mathbf{o}_i$ of the privacy-preserving processing of D_i by E_p. According to Theorem 2, the overall privacy guarantee of the outputs of $\mathcal{M}$ is equal to the sum of all the privacy budgets of the respective privacy mechanisms that compose $\mathcal{M}$, i.e., $\sum_{i \in T} \varepsilon_i$.

We assume that all the interactions between E_g and E_p are secure and private, and thus E_p is considered trusted and non-adversarial by E_g. Notice that, in a real life scenario, E_g and E_c might overlap with each other, i.e., data producers might be data consumers as well.

Privacy goal

We argue that in continuous user-generated data publishing, events are not equally significant in terms of privacy. We term a significant event—according to user- or data-related criteria—as a *landmark* event. The identification of landmark events can be performed manually or automatically, and is an orthogonal problem to ours. First, we consider the landmark timestamps, i.e., their position in time, non-sensitive and provided by the user as input along with the privacy budget ε. For example, events p_1, p_3, p_5, p_8 in Figure 4.1 are landmark events. In Definition 7, we formally introduce landmarks in the context of privacy-preserving data publishing.

Definition 7 (Landmark event)**.** *A landmark event is a significant—according to user- or data-related criteria—user-generated data item.*

Definition 8 extends the notion of neighboring data sets (see Section 2.2.5) to the context of landmarks.

Definition 8 (Landmark neighboring time series)**.** *Two time series of the same length, with common starting and ending timestamps, are* landmark neighboring *when their elements are pairwise, i.e., at the same timestamps, equal or neighboring*

and their neighboring elements are on common landmarks and/or at most on one regular event.

In Definition 9, we proceed to propose *landmark privacy*, a configurable variation of differential privacy for time series with significant events.

Definition 9 (Landmark privacy)**.** *Let $\mathcal{M}$ be a privacy mechanism with range $\mathcal{O}$ and domain $\mathcal{S}_T$ being the set of all time series with length $|T|$, where T is a sequence of timestamps. $\mathcal{M}$ satisfies landmark ε-differential privacy (or, simply, landmark privacy) if for all sets $O \subseteq \mathcal{O}$, and for every pair of landmark-neighboring time series S_T, S'_T, it holds that*

$$Pr[\mathcal{M}(S_T) \in O] \leq e^{\varepsilon} Pr[\mathcal{M}(S'_T) \in O]$$

User-level privacy can achieve landmark privacy, but it over-perturbs the final data by not distinguishing between landmark and regular events. Theorem 7 states how to achieve the desired privacy goal for the landmarks and any event, i.e., a total budget less than ε, and at the same time provide better utility overall.

Theorem 7 (Landmark privacy)**.** *Let $\mathcal{M}$ be a mechanism with input a time series S_T, where T is the set of the involved timestamps, and $L \subseteq T$ be the set of landmark timestamps. $\mathcal{M}$ is decomposed to ε-differential private sub-mechanisms $\mathcal{M}_t$, for every $t \in T$, which apply independent randomness to the event at t. Then, given a privacy budget ε, $\mathcal{M}$ satisfies* (ε, L)-landmark privacy *if for any t it holds that*

$$\sum_{i \in L \cup \{t\}} \varepsilon_i \leq \varepsilon$$

Proof. All mechanisms use independent randomness, and therefore for a time series $S_T = (D_i)_{i \in T}$ and outputs $(\boldsymbol{o}_i)_{i \in T} \in O \subseteq \mathcal{O}$ it holds that

$$Pr[\mathcal{M}(S_T) = (\boldsymbol{o}_i)_{i \in T}] = \prod_{i \in T} Pr[\mathcal{M}_i(D_i) = \boldsymbol{o}_i]$$

Likewise, for any landmark-neighboring time series S'_T of S_T with the same outputs $(\boldsymbol{o}_i)_{i \in T} \in O \subseteq \mathcal{O}$

$$Pr[\mathcal{M}(S'_T) = (\boldsymbol{o}_i)_{i \in T}] = \prod_{i \in T} Pr[\mathcal{M}_i(D'_i) = \boldsymbol{o}_i]$$

According to Definition 8, there exists $L \cup \{t\} \subseteq T$ such that $D_i = D'_i$ for $i \in L \cup \{t\}$. Thus, we get

$$\frac{Pr[\mathcal{M}(S_T) = (\boldsymbol{o}_i)_{i \in T}]}{Pr[\mathcal{M}(S'_T) = (\boldsymbol{o}_i)_{i \in T}]} = \prod_{i \in L \cup \{t\}} \frac{Pr[\mathcal{M}_i(D_i) = \boldsymbol{o}_i]}{Pr[\mathcal{M}_i(D'_i) = \boldsymbol{o}_i]}$$

D_i and D'_i are neighboring for $i \in L \cup \{t\}$. $\mathcal{M}_i$ is differential private and from Definition 2 we get that $\frac{Pr[\mathcal{M}_i(D_i)=\boldsymbol{o}_i]}{Pr[\mathcal{M}_i(D'_i)=\boldsymbol{o}_i]} \leq e^{\varepsilon_i}$. Hence, we can write

$$\frac{Pr[\mathcal{M}(S_T) = (\boldsymbol{o}_i)_{i\in T}]}{Pr[\mathcal{M}(S'_T) = (\boldsymbol{o}_i)_{i\in T}]} \leq \prod_{i\in L\cup\{t\}} e^{\varepsilon_i} = e^{\sum_{i\in L\cup\{t\}}\varepsilon_i}$$

For any $O \in \mathcal{O}$ we get $\frac{Pr[\mathcal{M}(S_T)\in O]}{Pr[\mathcal{M}(S'_T)\in O]} \leq e^{\sum_{i\in L\cup\{t\}}\varepsilon_i}$. If the formula of Theorem 7 holds, then we get $\frac{Pr[\mathcal{M}(S_T)\in O]}{Pr[\mathcal{M}(S'_T)\in O]} \leq e^{\varepsilon}$. Due to Definition 9 this concludes our proof. □

4.1.3 Achieving landmark privacy

In this section, we propose the methodology for achieving landmark privacy.

Landmark privacy mechanisms

`Uniform` Figure 4.3 shows the implementation of the baseline landmark privacy scheme for Example 4.1.1 which distributes uniformly the available privacy budget ε. In this case, it is enough to distribute at each timestamp the total privacy budget divided by the number of timestamps corresponding to landmarks, plus one, i.e., $\frac{\varepsilon}{|L|+1}$. Consequently, at each timestamp we protect every landmark, while reserving a part of ε for the current timestamp.

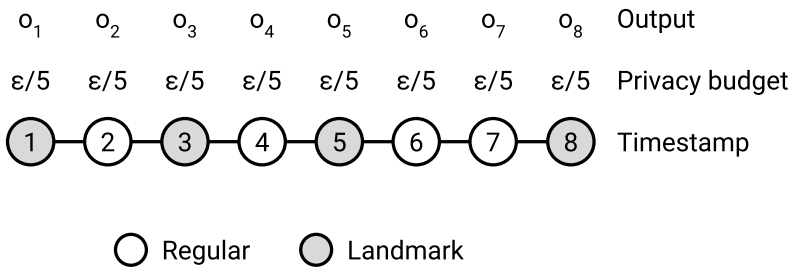


Figure 4.3: The `Uniform` application scenario of landmark privacy.

`Skip` One might argue that we could skip the landmark data releases as we demonstrate in Figure 4.4, by republishing previous, regular event releases. This would result in preserving all of the available privacy budget for regular events, equivalently to event-level protection, i.e., $\varepsilon_i = \varepsilon$, $\forall i \in T \setminus L$.

In practice, however, this approach can eventually pose arbitrary privacy risks, especially when dealing with geotagged data. Particularly, sporadic location data publishing or misapplying location cloaking could result in areas with sparse data

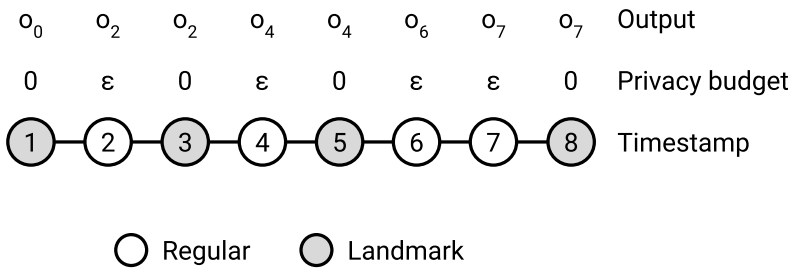


Figure 4.4: Application scenario of the `Skip` landmark privacy scheme.

points, indicating privacy-sensitive locations [GKdPC10, Rus18]. We study this problem and investigate possible solutions in Section 4.2.3.

`Adaptive` Next, we propose an adaptive privacy scheme (Figure 4.5) that accounts for changes in the input data by exploiting the post-processing property of differential privacy (Theorem 6).

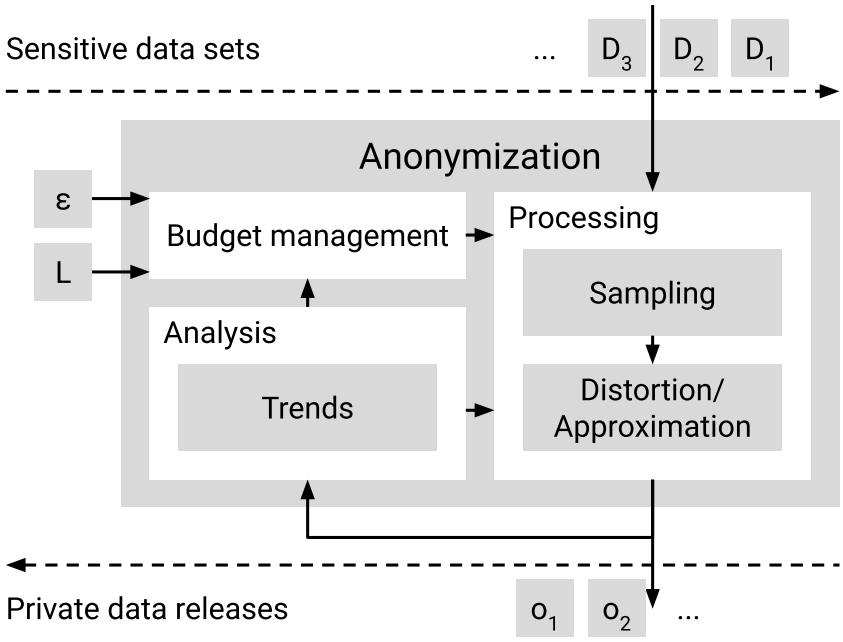


Figure 4.5: Concept of `Adaptive` landmark privacy.

Initially, its budget management component reserves uniformly the available privacy budget ε for each future release $\mathbf{o}$. At each timestamp, the processing component decides to either sample from the time series the current input and publish it with noise or release an approximation based on previous releases. In

the case when it publishes with noise the original data, the analysis component estimates the data trends by calculating the difference between the current and the previous releases and compares the difference with the scale of the perturbation, i.e., $\frac{\Delta f}{\varepsilon}$ [KPXP14]. The outcome of this comparison determines the adaptation of the sampling rate of the processing component for the next events: if the difference is greater it means that the data trends are evolving, and therefore it must increase the sampling rate. In the case when the mechanism approximates a landmark (but not a regular timestamp), the budget management component distributes the reserved privacy budget to the next timestamps. Due to the post-processing property of differential privacy (Theorem 6), the analysis component does not consume any privacy budget allowing for better final data utility.

Landmark privacy under temporal correlation

From the discussion so far, it is evident that for the budget distribution it is not the positions, but rather the number of the landmarks that matters. However, this is not the case under the presence of temporal correlation.

The Hidden Markov Model scheme (as used in [CYXX18]) stipulates two important independence properties: (i) the future (or past) depends on the past (or future) via the present, and (ii) the current observation is independent of the rest given the current state. Hence, there is independence between an observation at a specific timestamp and previous/next data sets under the presence of the current input data set. Intuitively, knowing the data set at timestamp t stops the propagation of the Markov chain towards the next or previous timestamps in the time series.

In Section 2.2.5 we showed that the temporal privacy loss α_t at a timestamp t is calculated as the sum of the backward and forward privacy loss, α_t^B and α_t^F, minus the privacy budget ε_t, to account for the extra privacy loss due to previous and next releases $\boldsymbol{o}$ of $\mathcal{M}$ under temporal correlation. By Theorem 7, at every timestamp t we consider the data at t and at the landmark timestamps L. When sequentially composing the data releases for each timestamp i in $L \cup \{t\}$ we consider the previous releases in the whole time series until the timestamp i^- that is exactly before i in the ordered $L \cup \{t\}$, and the next data releases in the whole time series until the timestamp i^+ that is exactly after i in the ordered $L \cup \{t\}$. Figure 4.6 illustrates i^- and i^+ in Example 4.0.1).

Therefore, in Definition 10, we formulate the landmark temporal privacy loss as follows.

Definition 10 (Landmark temporal privacy loss)**.** *Given a landmark set L in a set of timestamps T, the potential overall temporal privacy loss of a privacy*

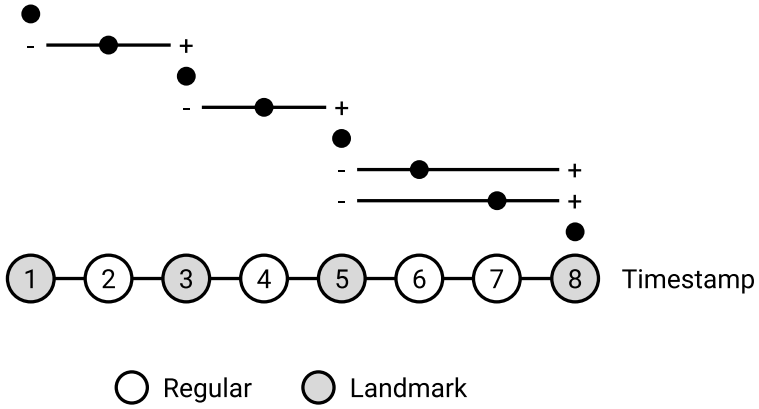


Figure 4.6: The timestamps exactly before (−) and after (+) every timestamp, where that is applicable, for the calculation of the temporal privacy loss.

mechanism $\mathcal{M}$ *at any timestamp in* $L \cup \{t\}$ *is*

$$\sum_{i \in L \cup \{t\}} \alpha_i$$

where for $i^-, i^+ \in L \cup \{t\}$ *being the timestamps exactly before and after* i, α_i *is equal to*

$$\underbrace{\ln \frac{\Pr[(\boldsymbol{o})_{i \in [i^- + 1, i]} | D_i]}{\Pr[(\boldsymbol{o})_{i \in [i^- + 1, i]} | D'_i]}}_{\alpha_i^B} + \underbrace{\ln \frac{\Pr[(\boldsymbol{o})_{i \in [i, i^+ - 1]} | D_i]}{\Pr[(\boldsymbol{o})_{i \in [i, i^+ - 1]} | D'_i]}}_{\alpha_i^F} - \underbrace{\ln \frac{\Pr[\boldsymbol{o}_i | D_i]}{\Pr[\boldsymbol{o}_i | D'_i]}}_{\varepsilon_i} \tag{4.1}$$

As presented in [CYXX18], the temporal privacy loss of a time series (without landmarks) can be bounded by a given privacy budget ε. Intuitively, by Equation 4.1 the temporal privacy loss incurred when considering landmarks is less than the temporal loss in the case without the knowledge of the landmarks. Thus, the temporal privacy loss in landmark privacy can be also bounded by ε.

4.2 Selection of events

In Section 4.1, we introduced the notion of landmark events in privacy-preserving time series publishing. The differentiation among regular and landmark events stipulates a privacy budget allocation that deviates from the application of existing differential privacy protection levels. Based on this novel event categorization, we designed three schemes (Section 4.1.3) that achieve landmark privacy. For this,

we assumed that the timestamps in the landmark set L are not privacy-sensitive, and therefore we used them in our models as they were.

This may pose a direct or indirect privacy risk to the users. For the former, we consider the case where we desire to publish L as complimentary information to the release of the event values. For the latter, a potentially adversarial data analyst may infer L by observing the values of the privacy budget, which is usually an inseparable attribute of the data release as an indicator of the privacy guarantee to the users and as an estimate of the data utility to the analysts. Hence, in both cases, a user-defined L, which is supposed to facilitate the configurable privacy protection of the user, could end up posing a privacy risk to them.

In Example 4.2.1, we demonstrate the extreme case of the application of the `Skip` landmark privacy scheme from Figure 4.4, where we approximate landmarks with the latest data release and invest all of the available privacy budget to regular events.

Example 4.2.1. *Figure 4.7 shows the privacy risk that the application of a landmark privacy scheme that nullifies or approximates outputs, similar to* ***`Skip`****, might cause. We point out in red the details that might cause indirect information inference. In this extreme case, the minimization of the privacy budget in combination with nullifying the output (either by not publishing or by adding a lot of noise) or approximating the current output with previously released outputs might hint to any adversary that the current event is a landmark.*

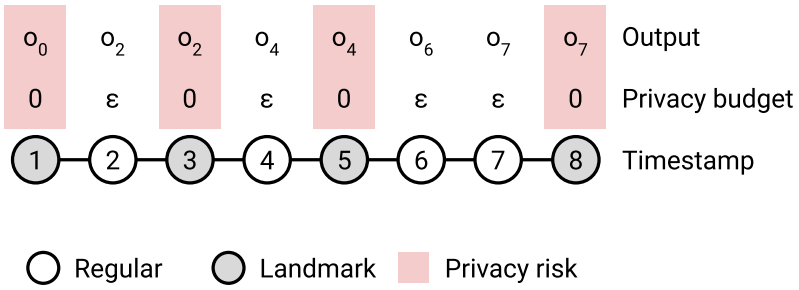


Figure 4.7: The privacy risk (highlighted in red) that the application of the landmark privacy `Skip` scheme might pose.

Apart from the privacy budget that we invested at landmarks, we can observe a pattern for the budgets at regular events as well. Therefore, an adversary who observes the values of the privacy budget can easily infer not only the number but also the exact temporal position of the landmarks.

4.2.1 Contribution

In this section, we extend the threat model, that we defined in Section 4.1.2, by taking into account the landmark set L as well. Simply put, we answer the question *'How can we protect the fact that we care more about certain events?'*. We design an additional differential privacy mechanism, based on the exponential mechanism (see Section 2.2.5 for more details), that we can easily plug-in to the existing landmark privacy mechanisms that we presented in Section 4.1.3.

4.2.2 Problem definition

The problem setting is similar to the one that we described in detail in Section 4.1.2. The main difference in this case lies in our threat model where we consider, in addition to the values of the regular and landmark events, the landmark timestamps L as privacy-sensitive as well.

One approach would be to utilize the randomized response (described in detail in Section 2.2.5) and randomize the answer to the question *'Is the current event a landmark?'* for every timestamp in $T \supseteq L$ of the time series S_T. However, this could result in a new landmark set L' that does not include all (or even any) of the timestamps in L. This contradicts the main idea of landmark privacy, i.e., take into account all landmarks at every timestamp.

4.2.3 Protecting landmarks

The main idea of the privacy-preserving dummy landmark selection module is to privately select extra landmark event timestamps, i.e., dummy landmarks, from the set of timestamps $T \setminus L$ of the time series S_T and add them to the original landmark set L. Selecting extra events, on top of the actual landmarks, as dummy landmarks, can render the actual ones indistinguishable. The goal is to create a new set L' such that $L \subset L' \subseteq T$.

First, we generate a set of dummy landmark set options by adding regular event timestamps from $T \setminus L$ to L (Section 4.2.3). Then, we utilize the exponential mechanism, with a utility function that calculates an indicator for each of the options in the set, based on how much it differs from the original landmark set L, and randomly select one of the options (Section 17). This process provides an extra layer of privacy protection to landmarks, and thus allows the processing, and thereafter releasing, of landmark timestamps.

Dummy landmark selection

Algorithms 1 and 2 approach this problem with an optimal and heuristic methodology, respectively. Function `evalSeq` evaluates the result of the union of L and

a timestamp combination from $T \setminus L$ by, e.g., estimating the standard deviation of all the distances from the previous/next landmark. `getOpts` returns all the possible *valid* sets of combinations `opt` such that larger options contain all of the timestamps that are present in smaller ones. Each combination contains a set of timestamps with sizes $|L|+1, |L|+2, \ldots, |T|$, where each one of them is a combination of L with $x \in [1, |T|-|L|]$ timestamps from T.

`Optimal` The `Optimal` algorithm (Algorithm 1) generates every possible combination (options) of landmark sets L' containing one set from every possible size, i.e, $|L|+1, |L|+2, \ldots, |T|$. Each L' contains the original landmarks along with timestamps of regular events from $T \setminus L$ (dummy landmarks). Then, it evaluates each option by comparing each of its sets with the original landmark set L and estimating an overall similarity score for each option (Lines 4–11). We discuss possible utility score functions later on in Section 17. It finds the option that is the most *similar* to the original (Lines 7-11), i.e., the option that has an evaluation that differs the least from that of the sequence T with landmarks L. The goal of this process is to select the option that contains the combination of dummy landmark sets that achieve the best score.

Algorithm 1: `Optimal` dummy landmark set options generation

Data: the time series timestamps T, the landmark set L
Output: the selected landmark set options opts

```
1  evalOrig ← evalSeq(T, ∅, L)
2  diffMin ← ∞
3  opts ← []
4  foreach opt ∈ getOpts(T, L) do
5  |  evalCur ← 0
6  |  foreach opt_i ∈ opt do
7  |  |  evalCur ← evalCur + evalSeq(T, opt_i, L)/#opt
8  |  diffCur ← |evalCur − evalOrig|
9  |  if diffCur < diffMin then
10 |  |  diffMin ← diffCur
11 |  |  opts ← opt
12 return opts
```

Algorithm 1 guarantees to return the optimal option with regard to the original set L. However, it is rather costly in terms of complexity. In more detail, given $|T \setminus L|$ regular events and a combination of size r, it requires $O(C(|T \setminus L|, r) + 2^{C(|T \setminus L|, r)})$ time and $O(r * C(|T \setminus L|, r))$ space. Next, we present a `Heuristic` solution with improved time and space requirements.

`Heuristic` The `Heuristic` algorithm (Algorithm 2) follows an incremental methodology and at each step it selects a new timestamp, corresponding to a regular event from $T \setminus L'$. In this case, the elements of L' at each step differ by one from the one that the algorithm selected in the previous step. Similar to the `Optimal`, it selects a new set based on a predefined similarity metric until it selects a set that is equal to the size of the series of events, i.e., $L' = T$.

Algorithm 2: `Heuristic` dummy landmark set options generation

Data: the time series timestamps T, the landmark set L
Output: the selected landmark set options opts

```
1  evalOrig ← evalSeq(T, ∅, L)
2  opts ← []
3  L' ← L
4  while L' ≠ T do
5      diffMin ← ∞
6      optim_i ← Null
7      foreach reg ∈ T \ L' do
8          evalCur ← evalSeq(T, reg, L')
9          diffCur ← |evalCur − evalOrig|
10         if diffCur < diffMin then
11             diffMin ← diffCur
12             optim_i ← reg
13     L'.add(optim_i)
14     opts.append(L' \ L)
15 return opts
```

Similar to Algorithm 1, it selects new options based on a predefined metric (Lines 8-12). This process (Lines 4-14) goes on until we select a set that is equal to the size of the series of events, i.e., $L' = T$. In terms of complexity, given $|T \setminus L|$ regular events, the `Heuristic` requires $O(|T \setminus L|^2)$ time and space. Note that the reverse process, i.e., starting with T landmarks and removing until $|L'| = |L| + 1$, performs similarly.

`Partitioned` We improve the complexity of the `Heuristic` algorithm by partitioning the landmark timestamp sequence L. The novelty of this algorithm lies in the fact that it deals with the event series as a histogram which allows it to take advantage of its relevant features and methodology. Particularly, it uses the Freedman-Diaconis rule, which is resilient to outliers and takes into account the data variability and data size [MI15], and generates a histogram from the landmark set L. This way, it achieves an improved complexity, compared to the `Heuristic`,

that is dependent on the histogram's bin size. Algorithm 3 demonstrates the overall process.

Algorithm 3: `Partitioned` dummy landmark set options generation

Data: the time series timestamps T, the landmark set L
Output: the selected landmark set options opts

```
1  hist, h ← getHist(T, L)
2  histCur ← hist
3  opts ← []
4  while sum(histCur) ≠ len(T) do
5      diffMin ← ∞
6      opt ← histCur
7      foreach h_i in histCur do
8          if h_i + 1 ≤ h then
9              histTmp ← histCur
10             histTmp[i] ← histTmp[i] + 1
11             diffCur ← getDiff(hist, histTmp)
12             if diffCur < diffMin then
13                 diffMin ← diffCur
14                 opt ← histTmp
15     histCur ← opt
16     opts ← opt
17 return opts
```

Function `getHist` generates a histogram with bins of size h for a given time series timestamps T and landmark set L. For every new histogram version, the `getDiff` function (Line 11) finds the difference from the original histogram; for this operation it utilizes the Euclidean distance (see Section 5.3.1 for more details). In Lines 7-14, the algorithm checks every histogram version by incrementing each bin by 1 and comparing it to the original (Line 12). In the end, it returns opts which contains all the versions of hist that are closest to the original hist for all possible bin sizes of hist.

Privacy-preserving option selection

The algorithms that we presented in Section 4.2.3 return a set of possible versions of the original landmark set L by adding extra timestamps in it from the series of events at timestamps $T \setminus L$. In the next step, we randomly select a set by utilizing the exponential mechanism (Section 2.2.5). For this procedure, we allocate a small fraction of the available privacy budget, i.e., 1% or even less (see Section 5.3.2

for more details), which adds up to that of the publishing scheme according to Theorem 2.

Utility score function Prior to selecting a landmark timestamp set including the original along with dummy landmarks, the exponential mechanism evaluates each set using a utility score function. We present here two ways of doing so.

One way to evaluate each set is by taking into account the temporal position of the events in the sequence. Events that occur at recent timestamps are more likely to reveal sensitive information regarding the users involved [KPXP14]. Hence, indicating the existence of dummy landmarks nearby actual landmarks can increase the adversarial confidence regarding the location of the latter within a series of events. In other words, sets with dummy landmarks with less average temporal distance from actual landmarks achieve better utility scores.

Another approach for the utility score function is to consider the number of events in each set. Sets with more dummy landmarks may render actual landmarks more indistinguishable, and therefore provide less utility. Consequently, more dummy landmarks lead to distributing the privacy budget to more events, and therefore leading to more robust overall privacy protection.

Option release In the last step, the privacy-preserving dummy landmark selection module releases a new landmark set (including the original landmarks along with the dummy ones) from the options that were generated in the previous step, by utilizing the exponential mechanism.

The options generated by the `Optimal` and `Heuristic` algorithms contain actual timestamps that can be utilized directly by the landmark privacy schemes that we presented in Section 4.1.3. However, the `Partitioned` algorithm returns histograms instead of timestamps. Therefore, we need to process the result of the exponential mechanism further by sampling without replacement from the set $T \setminus L$ according to the selected histogram's probability density function.

4.3 Summary

In this chapter, we presented *landmark privacy* for privacy-preserving time series publishing, which allows for the protection of significant events while improving the utility of the final result compared to user-level differential privacy. We proposed three schemes for landmark privacy, and quantified the privacy loss under temporal correlation. Furthermore, we designed a module to enhance our privacy notion by protecting the actual timestamps of the landmarks. We differ the experimental evaluation of our methodology to Chapter 5 we experiment with real and

synthetic data sets to demonstrate the applicability of the landmark privacy models by themselves (Section 5.3) and in combination with the landmark selection component (Section 5.2).

Chapter 5

Evaluation

In this chapter, we present the experiments that we performed in order to evaluate landmark privacy (Chapter 4) on real and synthetic data sets. Section 5.1 contains all the details regarding the data sets the we used for our experiments along with the system configurations. Section 5.2 evaluates the data utility of the landmark privacy schemes that we designed in Section 4.1 in comparison to user- and event-level, and investigates the behavior of the privacy loss under temporal correlation for different distributions of landmarks. Section 5.3 justifies our decisions while designing the privacy-preserving dummy landmark selection module in Section 4.2 and the data utility impact of the latter. Finally, Section 5.4 concludes this chapter by summarizing the main results derived from the experiments.

5.1 Setting, configurations, and data sets

In this section we list all the relevant details regarding the evaluation setting (Section 5.1.1), and we present the real and synthetic data sets that we used (Section 5.1.2) along with the corresponding configurations (Section 5.1.3).

5.1.1 Machine setup

We implemented our experiments[1] in Python 3.9.7 and executed them on a machine with an Intel i7-6700HQ at 3.5GHz CPU and 16GB RAM, running Manjaro

This chapter will appear in the proceedings of the 12th ACM conference on Data and Application Security and Privacy [KTK22].

[1]Source code available at `https://git.delkappa.com/manos/the-last-thing`

Linux 21.1.5. We repeated each experiment 100 times and we report the mean over these iterations.

5.1.2 Data sets

We performed experiments on real (Section 5.1.2) and synthetic data sets (Section 5.1.2).

Real data sets

For uniformity and in order to be consistent, we sample from each of the following data sets the first $1,000$ entries that satisfy the configuration criteria that we discuss in detail in Section 5.1.3.

Copenhagen [SSLL19] data set was collected via the smartphone devices of 851 university students over a period of 4 weeks as part of the Copenhagen Networks Study. Each device was configured to be discoverable by and to discover nearby Bluetooth devices every 5 minutes. Upon discovery, each device registers (i) the timestamp in seconds, (ii) the device's unique identifier, (iii) the unique identifier of the device that it discovered (-1 when no device was found or -2 for any non-participating device), and (iv) the Received Signal Strength Indicator (RSSI) in dBm. Half of the devices have registered data at at least 81% of the possible timestamps. 3 devices (449, 550, 689) satisfy our configuration criteria (Section 5.1.3) within their first $1,000$ entries. From those 3 devices, we picked the first one, i.e., device with identifier '449', and utilized its $1,000$ first entries out of $12,167$ unique valid contacts.

HUE [Mak18] contains the hourly energy consumption data of 22 residential customers of BCHydro, a provincial power utility in British Columbia. The measurements for each residence are saved individually and each measurement contains (i) the date (YYYY-MM-DD), (ii) the hour, and (iii) the energy consumption in kWh. In our experiments, we used the first residence, i.e., residence with identifier '1', that satisfies our configuration criteria (Section 5.1.3) within its first $1,000$ entries. In those entries, out of a total of $29,231$ measurements, we estimated an average energy consumption equal to 0.88kWh and a value range within [0.28, 4.45].

T-drive [YZZ+10] consists of 15 million GPS data points of the trajectories of $10,357$ taxis in Beijing, spanning a period of 1 week and a total distance of 9 million kilometers. The taxis reported their location data on average every 177 seconds and 623 meters approximately. Each vehicle registers (i) the taxi unique

identifier, (ii) the timestamp (YYYY-MM-DD HH:MM:SS), (iii) the longitude, and (iv) the latitude. These measurements are stored individually per vehicle. We sampled the first 1,000 data items of the taxi with identifier '2', which satisfied our configuration criteria (Section 5.1.3).

Synthetic data sets

We generated synthetic time series of length equal to 100 timestamps, for which we varied the number and distribution of landmarks. In this way, we have a controlled data set that we can use to study the behavior of our proposal. We take into account only the temporal order of the points and the position of regular and landmark events within the time series. In Section 5.1.3, we explain in more detail our configuration criteria.

5.1.3 Configurations

We vary the landmark percentage (Section 5.1.3), i.e., the ratio of timestamps that we attribute to landmarks and regular events, in order to explore the behavior of our methodology in all possible scenarios. For each data set, we implement a privacy mechanism that injects noise related to the type of its attribute values and we tune the parameters of each mechanism accordingly (Section 5.1.3). Last but not least, we explain how we generate synthetic data sets with various degrees of temporal correlation so as to observe the impact on the overall privacy loss (Section 5.1.3).

Landmark percentage

In the Copenhagen data set, a landmark represents a timestamp when a specific contact device is registered. After identifying the unique contacts within the sample, we achieve each desired landmarks to regular events ratio by considering a list that contains a part of these contact devices. In more detail, we achieve 0% landmarks by considering an empty list of contact devices, 20% by extending the list with [3, 6, 11, 12, 25, 29, 36, 39, 41, 46, 47, 50, 52, 56, 57, 61, 63, 78, 80], 40% with [81, 88, 90, 97, 101, 128, 130, 131, 137, 145, 146, 148, 151, 158, 166, 175, 176], 60% with [181, 182, 192, 195, 196, 201, 203, 207, 221, 230, 235, 237, 239, 241, 254], 80% with [260, 282, 287, 289, 290, 291, 308, 311, 318, 323, 324, 330, 334, 335, 344, 350, 353, 355, 357, 358, 361, 363], and 100% by including all of the possible contacts.

In HUE, we consider as landmarks the events that have energy consumption values below a certain threshold. That is, we get 0%, 20% 40%, 60%, 80%, and 100%

landmarks by setting the energy consumption threshold to 0.28kWh, 1.12kWh, 0.88kWh, 0.68kWh, 0.54kWh, and 4.45kWh respectively.

In T-drive, a landmark represents a location where a vehicle spend some time. We achieved the desired landmark percentages by utilizing the method of Li et al. [LZX+08] for detecting stay points in trajectory data. In more detail, the algorithm checks for each data item if each subsequent item is within a given distance threshold Δl and measures the time period Δt between the present point and the last subsequent point. After analyzing the data and experimenting with different pairs of distance and time period, we achieve 0%, 20% 40%, 60%, 80%, and 100% landmarks by setting the (Δl in meters, Δt in minutes) pairs input to the stay point discovery method as [(0, 1000), (2095, 30), (2790, 30), (3590, 30), (4825, 30), (10350, 30)].

We generated synthetic data with *skewed* (the landmarks are distributed towards the beginning/end of the series), *symmetric* (in the middle), *bimodal* (both end and beginning), and *uniform* (all over the time series) landmark distributions. In order to get landmark sets with the above distribution features, we generate probability distributions with restricted domain to the beginning and end of the time series, and sample from them, without replacement, the desired number of points. For each case, we place the location (centre) of the distribution accordingly. That is, for symmetric we put the location in the middle of the time series and for left/right skewed to the right/left. For bimodal we combine two mirrored skewed distributions. Finally, for the uniform distribution we distribute the landmarks randomly throughout the time series. For consistency, we calculate the scale parameter of the corresponding distribution depending on the length of the time series by setting it equal to the series' length over a constant.

Privacy parameters

For all of the real data sets, we implement ε-differential privacy by selecting a mechanism, from those that we described in Section 2.2.5, that is best suited for the type of its sensitive attributes. To perturb the contact tracing data of the Copenhagen data set, we utilize the *random response* technique [WBLJ17] and we report truthfully at each timestamp, with probability $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}$, whether the current contact is a landmark or not. We randomize the energy consumption in HUE with the Laplace mechanism [DR+14]. For T-drive, we perturb the location data with noise that we sample from a Planar Laplace distribution [ABCP13].

We set the privacy budget $\varepsilon = 1$ for all of our experiments and, for simplicity, we assume that for every query sensitivity it holds that $\Delta f = 1$. For the experiments that we performed on the synthetic data sets, the original values to be released are not relevant to what we want to observe, and thus we ignore them.

Temporal correlation

Despite the inherent presence of temporal correlation in time series, it is challenging to correctly discover and quantify it. For this reason, and in order to create a more controlled environment for our experiments, we chose to conduct tests relevant to temporal correlation using synthetic data sets. We model the temporal correlation in the synthetic data as a *stochastic matrix* P, using a *Markov Chain* [Gag17]. P is an $n \times n$ matrix, where the element P_{ij} represents the transition probability from a state i to another state j, $\forall\, i,\, j \leq n$. It holds that the elements of every row j of P sum up to 1. We follow the *Laplacian smoothing* technique [SCOL+04], as utilized in [CYXX18], to generate the matrix P with a degree of temporal correlation $s > 0$ equal to

$$\frac{(I_n)_{ij} + s}{\sum_{k=1}^{n}((I_n)_{jk} + s)}$$

where I_n is an *identity matrix* of size n. The value of s is comparable only for stochastic matrices of the same size and dictates the strength of the correlation; the lower its value, the stronger the correlation degree. In our experiments, for simplicity, we set $n = 2$ and we investigate the effect of *weak* ($s = 1$), *moderate* ($s = 0.1$), and *strong* ($s = 0.01$) temporal correlation degree on the overall privacy loss.

5.2 Landmark events

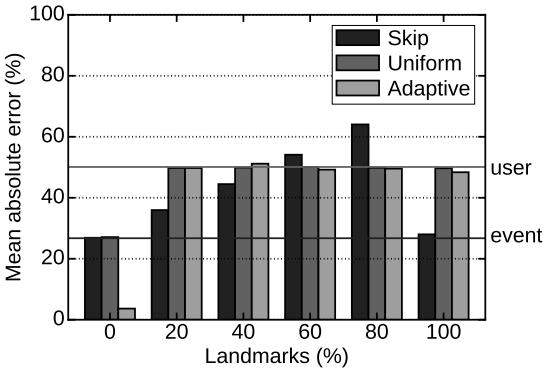
In this section, we present the experiments that we performed, to test the methodology that we presented in Section 4.1.3, on real and synthetic data sets.

With the experiments on the real data sets (Section 5.2.1), we show the performance in terms of data utility of our three landmark privacy schemes: `Skip`, `Uniform` and `Adaptive`. We define data utility as the mean absolute error introduced by the privacy mechanism. We compare with the event- and user-level differential privacy protection levels, and show that, in the general case, landmark privacy allows for better data utility than user-level differential privacy while balancing between the two protection levels.

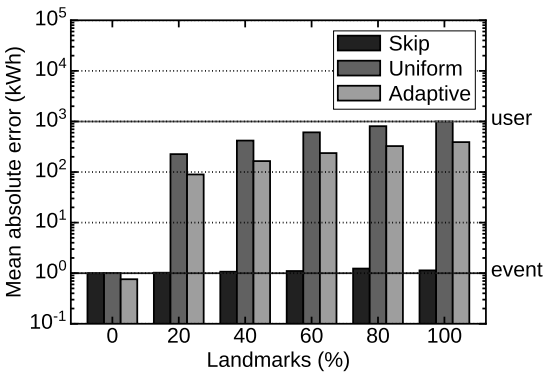
With the experiments on the synthetic data sets (Section 5.2.2) we show how the temporal privacy loss, i.e., the privacy budget ε with the extra privacy loss because of the temporal correlation, changes when tuning the size and statistical characteristics of the input landmark set L. We observe that a greater average landmark–regular event distance in a time series can result into greater temporal privacy loss under moderate and strong temporal correlation.

5.2.1 Landmark privacy schemes

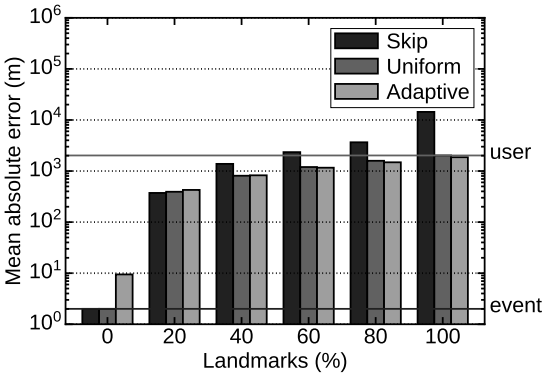
Figure 5.1 exhibits the performance of the three schemes, `Skip`, `Uniform`, and `Adaptive` applied on the three data sets that we study. Notice that, in the cases when we have 0% and 100% of the events being landmarks, we get the same behavior as in event- and user-level privacy respectively. This happens due to the fact that at each timestamp we take into account only the data items at the current timestamp and ignore the rest of the time series (event-level) when there are no landmarks. Whereas, when each timestamp corresponds to a landmark we consider and protect all the events throughout the entire series (user-level).



(a) Copenhagen



(b) HUE



(c) T-drive

Figure 5.1: The mean absolute error (a) as a percentage, (b) in kWh, and (c) in meters of the released data for different landmark percentages.

For the Copenhagen data set (Figure 5.1a), `Adaptive` has an overall consistent performance and works best for 60% and 80% landmarks. We notice that for 0% landmarks, it achieves better utility than the event-level protection due to the combination of more available privacy budget per timestamp (due to the absence

of landmarks) and its adaptive sampling methodology. `Skip` excels, compared to the others, at cases where it needs to approximate 20%, 40%, or 100% of the times. In general, we notice that, for this data set and due to the application of the random response technique, it is more beneficial to either invest more privacy budget per event or prefer approximation over introducing randomization.

The combination of the small range of measurements ([0.28, 4.45] with an average of 0.88kWh) in HUE (Figure 5.1b) and the large scale in the Laplace mechanism, allows for schemes that favor approximation over noise injection to achieve a better performance in terms of data utility. Hence, `Skip` achieves a constant low mean absolute error. Regardless, the `Adaptive` scheme performs by far better than `Uniform` and balances between event- and user-level protection for all landmark percentages.

In T-drive (Figure 5.1c), `Adaptive` outperforms `Uniform` by 10%–20% for all landmark percentages greater than 40% and `Skip` by more than 20%. The lower density (average distance of 623m) of the T-drive data set has a negative impact on the performance of `Skip` because republishing a previous perturbed value is now less accurate than perturbing the current location.

Principally, we can claim that the `Adaptive` is the most reliable and best performing scheme, if we take into consideration the drawbacks of the Skip mechanism, particularly in spatiotemporal data, e.g., sporadic location data publishing [GKdPC10, Rus18] or misapplying location cloaking [xss21], that could lead to the indication of privacy-sensitive attribute values. Moreover, implementing a more advanced and data-dependent sampling method that accounts for changes in the trends of the input data and adapts its rate accordingly, would result in a more effective budget allocation that would improve the performance of `Adaptive` in terms of data utility.

5.2.2 Temporal distance and correlation

As previously mentioned, temporal correlation is inherent in continuous publishing, and it is the cause of supplementary privacy loss in the case of privacy-preserving time series publishing. In this section, we are interested in studying the effect that the distance of the landmarks from every regular event has on the privacy loss caused under the presence of temporal correlation.

Figure 5.2 shows a comparison of the average temporal distance of the events from the previous/next landmark or the start/end of the time series for various distributions in our synthetic data. More specifically, we model the distance of an event as the count of the total number of events between itself and the nearest landmark or the time series edge.

We observe that the uniform and bimodal distributions tend to limit the regular event–landmark distance. This is due to the fact that the former scatters the

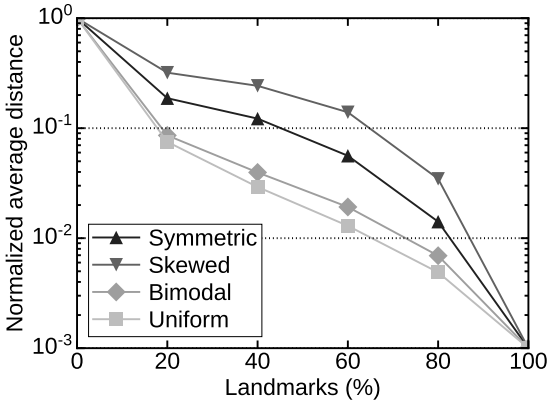
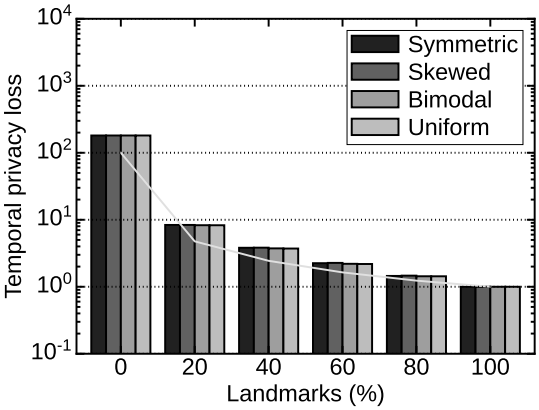


Figure 5.2: Average temporal distance of regular events from the landmarks for different landmark percentages within a time series in various landmark distributions.

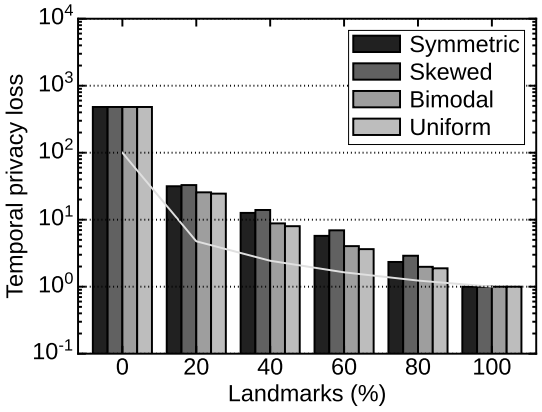
landmarks, while the latter distributes them on both edges, leaving a shorter space uninterrupted by landmarks. On the contrary, distributing the landmarks at one part of the sequence, as in skewed or symmetric, creates a wider space without landmarks. This study provides us with different distance settings that we are going to use in the subsequent temporal privacy loss study.

Figure 5.3 illustrates a comparison among the aforementioned distributions regarding the temporal privacy loss under (a) weak, (b) moderate, and (c) strong temporal correlation degrees. The line shows the overall privacy loss—for all cases of landmark distribution—without temporal correlation.

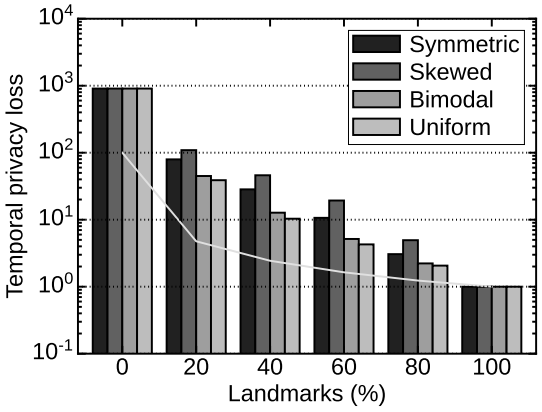
In combination with Figure 5.2, we conclude that a greater average landmark–regular event distance in a distribution can result into greater temporal privacy loss under moderate and strong temporal correlation. This is due to the fact that the backward/forward privacy loss accumulates more over time in wider spaces without landmarks (see Section 2.3). Furthermore, the behavior of the privacy loss is as expected regarding the temporal correlation degree: a stronger correlation degree generates higher privacy loss while widening the gap between the different distribution cases. On the contrary, a weaker correlation degree makes it harder to differentiate among the landmark distributions. The privacy loss under a weak correlation degree converge with all possible distributions for all landmark percentages.



(a) Weak correlation



(b) Moderate correlation



(c) Strong correlation

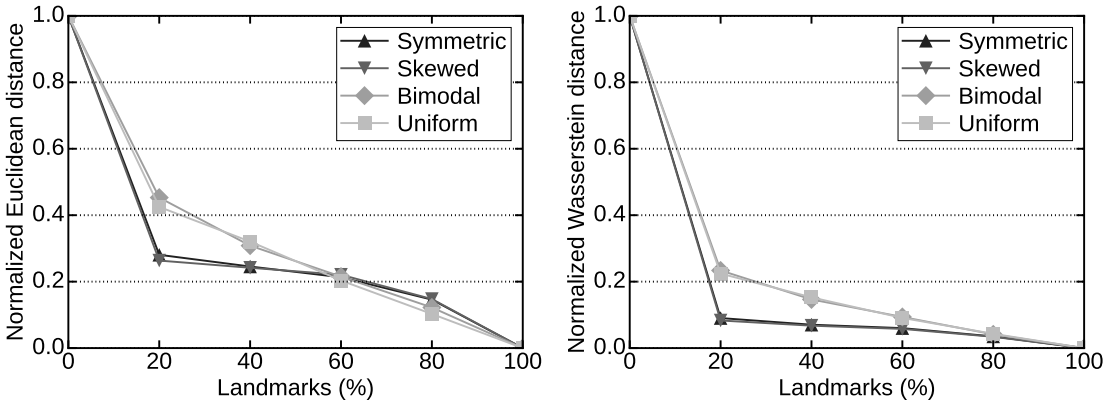
Figure 5.3: The temporal privacy loss for different landmark percentages and distributions under (a) weak, (b) moderate, and (c) strong degrees of temporal correlation. The line shows the overall privacy loss without temporal correlation.

5.3 Selection of landmarks

In this section, we present the experiments on the methodology for the dummy landmark selection presented in Section 4.2.3, on the real and synthetic data sets. Due to the high complexity of the `Optimal` and `Heuristic` algorithms, we choose to evaluate only the `Partitioned`, which is the optimized solution that we designed. With the experiments on the synthetic data sets (Section 5.3.1) we show the normalized Euclidean and Wasserstein distance metrics (not to be confused with the temporal distances in Figure 5.2) of the time series histograms for various distributions and landmark percentages. This allows us to justify our design decisions for our concept that we showcased in Section 4.2.3. With the experiments on the real data sets (Section 5.3.3), we show the performance in terms of utility of our three landmark schemes in combination with the privacy-preserving dummy landmark selection module, which enhances the privacy protection that our concept provides.

5.3.1 Dummy landmark selection utility metrics

Figure 5.4 demonstrates the normalized distance that we obtain when we utilize either (a) the Euclidean or (b) the Wasserstein distance metric to obtain a set of landmarks including regular events.



(a) Euclidean (b) Wasserstein

Figure 5.4: The normalized (a) Euclidean, and (b) Wasserstein distance of the generated landmark sets for different landmark percentages.

Comparing the results of the Euclidean distance in Figure 5.4a with those of the Wasserstein in Figure 5.4b we conclude that the Euclidean distance provides more consistent results for all possible distributions. The maximum difference per

landmark percentage is approximately 0.2 for the former and 0.15 for the latter between the bimodal and skewed landmark distributions. Overall, the Euclidean distance achieves a mean normalized distance of 0.3, while the Wasserstein distance a mean normalized distance that is equal to 0.2. Therefore, and by observing Figure 5.4, Wasserstein demonstrates a less consistent performance and less linear behavior among all possible landmark distributions. Thus, we choose to utilize the Euclidean distance metric for the implementation of the privacy-preserving dummy landmark selection module in Section 4.2.3.

5.3.2 Privacy budget tuning

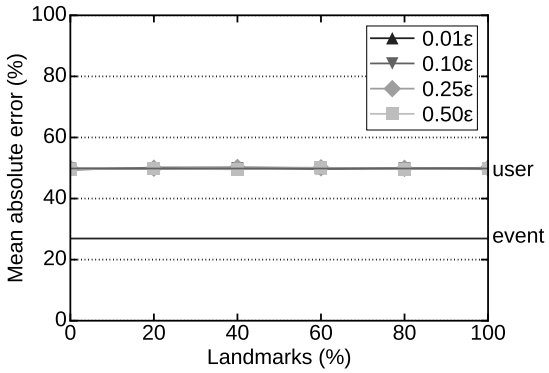
In Figure 5.5, we test the `Uniform` mechanism with real data by investing different ratios (1%, 10%, 25%, and 50%) of the available privacy budget ε in the dummy landmark selection module and the remaining in perturbing the original data values, in order to figure out the optimal ratio value. `Uniform` is our baseline implementation, and hence allows us to derive more accurate conclusions in this case. In general, we are expecting to observe that greater ratios will result in more accurate, i.e., smaller, landmark sets and less accurate values in the released data.

The application of the randomized response mechanism, in the Copenhagen data set (Figure 5.5a), is tolerant to the fluctuations of the privacy budget and maintains a relatively constant performance in terms of data utility. For HUE (Figure 5.5b) and T-drive (Figure 5.5c), we observe that our implementation performs better for lower ratios, e.g., 0.01, where we end up allocating the majority of the available privacy budget to the data release process instead of the dummy landmark selection module. The results of this experiment indicate that we can safely allocate the majority of ε to the data publishing process, and therefore achieve better data utility, while guaranteeing more robust privacy protection.

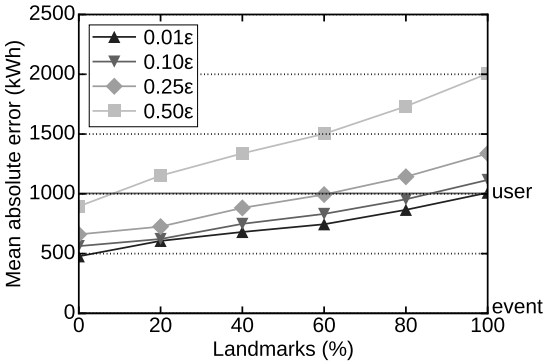
5.3.3 Privacy schemes and dummy landmark selection

Figure 5.6 exhibits the performance of `Skip`, `Uniform`, and `Adaptive` schemes (presented in detail in Section 4.1.3) in combination with the landmark selection module (Section 4.2.3).

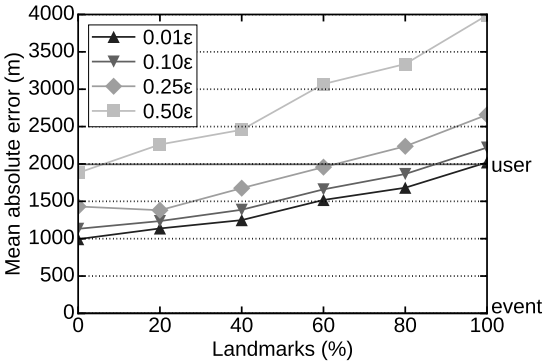
In comparison with the utility performance without the dummy landmark selection module (solid bars), we notice a slight deterioration for all three schemes (markers). This is natural since we allocated part of the available privacy budget to the privacy-preserving dummy landmark selection module, which in turn increased the number of landmarks, except for the case of 100% landmarks. Therefore, there is less privacy budget available for data publishing throughout the time series. `Skip` performs best in our experiments with HUE (Figure 5.6b), due to the low range in the energy consumption and the high scale of the Laplace noise



(a) Copenhagen

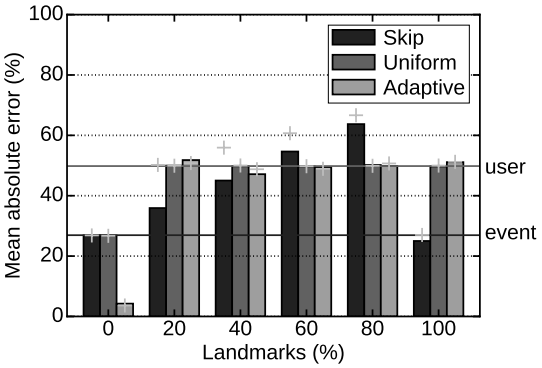


(b) HUE

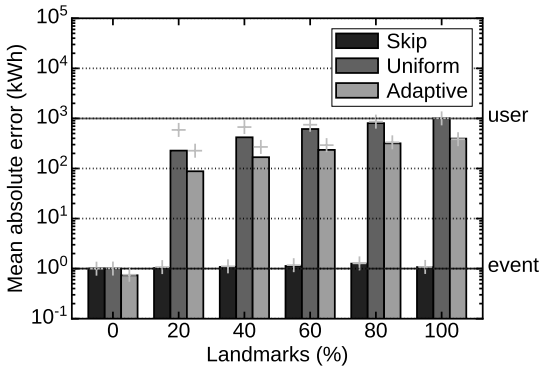


(c) T-drive

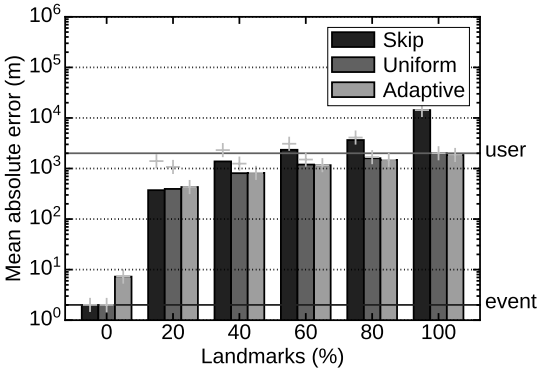
Figure 5.5: The mean absolute error (a) as a percentage, (b) in kWh, and (c) in meters of the released data for different landmark percentages. We apply the `Uniform` landmark privacy mechanism and vary the ratio of the privacy budget ε that we allocate to the dummy landmark selection module.



(a) Copenhagen



(b) HUE



(c) T-drive

Figure 5.6: The mean absolute error (a) as a percentage, (b) in kWh, and (c) in meters of the released data, for different landmark percentages from Figure 5.1. The markers indicate the corresponding measurements with the incorporation of the privacy-preserving landmark selection module.

that it avoids due to the employed approximation. However, for the Copenhagen data set (Figure 5.6a) and T-drive (Figure 5.6c), `Skip` attains high mean absolute error, which exposes no benefit with respect to user-level protection. Overall, `Adaptive` has a consistent performance in terms of utility for all of the data sets that we experimented with, and almost always outperforms the user-level privacy protection. Thus, `Adaptive` is selected as the best scheme to use in general.

5.4 Summary

In this chapter we presented the experimental evaluation of the landmark privacy schemes and the dummy landmark selection module, that we developed in Chapter 4, on real and synthetic data sets. The `Adaptive` scheme is the most reliable and best performing scheme, in terms of overall data utility, with minimal tuning across most of the cases. `Skip` performs optimally in data sets with a smaller target value range, where approximation fits best. The dummy landmark selection module introduces a reasonable data utility decline to all of our schemes; however, the `Adaptive` handles it well and bounds the data utility to higher levels compared to user-level protection. In terms of temporal correlation, we observe that under moderate and strong temporal correlation, a greater average regular–landmark event distance in a landmark distribution causes greater temporal privacy loss. Finally, the contribution of the landmark privacy on enhancing the data utility, while preserving ε-differential privacy, is demonstrated by the fact that the selected `Adaptive` scheme provides better data utility than the user-level privacy protection.

Chapter 6

Conclusion and future work

Continuous publishing of data, also known as time series, has found over the past decades several application domains, including healthcare, smart building, and traffic monitoring. In many cases, time series contain personal details, which are usually geotagged, and thus their processing entails privacy concerns. Several methods have been proposed in order to protect the privacy of individuals while processing their data, but cannot avoid to deteriorate arbitrarily the quality therein. Out of these methods, we distinguish differential privacy, which quantifies the balance between user protection and data utility by a factor ε.

In this thesis, we have concentrated on continuous user-generated data publishing. Particularly, we focused on providing differential privacy over finite time series while accounting for privacy-significant future and past events. We have studied the relevant literature with special emphasis on data correlation. Furthermore, we explored ways to provide configurable protection in such settings and developed relevant solutions.

Next, we summarize this thesis in the individual chapters by describing our contribution to the problems surrounding quality and privacy in continuous data publishing. Subsequently, we discuss interesting perspectives and open questions for future research.

6.1 Thesis summary

This thesis revolves around the topic of quality and privacy in user-generated Big Data, focusing on the problems regarding privacy-preserving continuous data publishing that we summarize below.

Survey on continuous data publishing We reviewed the existing literature regarding methods on privacy-preserving continuous data publishing, spanning the

past two decades, while elaborating on data correlation. Our contributions are:

- We categorized the works that we reviewed based on their input data in either *microdata* or *statistical data* and further separated each data category based on its observation span in *finite* and *infinite*.
- We identified the privacy protection algorithms and techniques that each work is using, focusing on feature like the privacy method, operation, attack, and protection level.
- We organized the reviewed literature in a tabular form to allow for a more efficient indexation of the related works, using a number of relevant features.

This work appeared in the special feature on Geospatial Privacy and Security of the 19th journal of Spatial Information Science [KTK19].

Configurable privacy protection for time series We presented (ε, L)-*landmark privacy*, a novel privacy notion that is based on differential privacy allowing for better data utility in the presence of significant events. Our contributions are:

- We introduced the notion of *landmark events* in privacy-preserving data publishing and differentiated events between regular and events that a user might consider more privacy-sensitive (*landmarks*).
- We designed and implemented three landmark privacy schemes for landmarks spanning a finite time series.
- We studied landmark privacy under temporal correlation, which is inherent in time series, and observed the effect of landmarks on the temporal privacy loss propagation.
- We designed an additional differential privacy mechanism, based on the exponential mechanism, for providing protection to the temporal position of the landmarks by generating dummy landmark set options.
- We experimentally evaluated our proposal on real and synthetic data sets, and compared landmark privacy schemes with event- and user-level privacy protection, for different landmark percentages. We showed that our methodology can provide adequate differential privacy guarantees while achieving better data utility than the user-level scheme.

This work will appear in the proceedings of the 12th ACM conference on Data and Application Security and Privacy [KTK22].

6.2 Perspectives

In this section, we outline possible perspectives to this thesis regarding the different topics that we addressed.

Diversification of event categories With the proposal of landmark privacy we introduced landmark events in privacy-preserving continuous data publishing. The categorization in regular and significant events enabled the development of a configurable differential privacy notion for time series. The variation of the existing event categories, e.g., weighted landmarks, or the introduction of new ones, would allow for an even more fine-grained configuration of privacy protection and the development of different versions of landmark privacy.

Global landmark privacy For now, we have applied landmark privacy in the local scheme and for microdata due to the advantages of the local scheme over the global as we discussed in detail in Section 2.1.2. Since we can easily adapt landmark privacy to the global processing and publishing scheme, it would be interesting to observe it in more diverse scenarios (including statistical data publishing) and develop suitable methodology.

Landmark privacy over infinite event sequences So far, we considered for our problem setting finite time series that are processed in batch mode. This was a decision that we made for reasons of clarity in order to facilitate a more straightforward definition of landmark privacy. In the future, fellow researchers can explore more dynamic scenarios where data are processed and published in streaming mode, which will lead to the adoption of time critical crowdsensing applications.

Landmark privacy and spatiotemporal continuity In mereology, the formal study on the relation between parts and the entities they form, it is generally held that the identity of an observable object depends to its *spatiotemporal continuity* [Wig67, Sca81, HC01], i.e., the property of well-behaved objects that alter their state in harmony with space and time. Considering events that span the entirety of the user-generated series of events thereof ensures the spatiotemporal continuity of the users. This way, it is possible to acquire more information regarding individuals' identities, and thus design privacy schemes that offer improved privacy and utility guarantees.

Consideration of other data correlation types In the current state of our work, we consider landmarks as one-dimensional elements in our problem setting.

Consequently, we have explored landmark privacy under temporal correlation and examined the behavior of temporal privacy loss for different landmark percentages and distributions. Accounting for other possible dimensions, e.g., location, can introduce more aspects to the current use case of landmark privacy. Indicatively, as we have extensively studied in Section 2.3, there are many types of data correlation in time series to further research in the context of landmark privacy.

Incorporation of machine learning Until now, we consider the landmark discovery and selection process orthogonal to our work. In the future, we aim to work on automatically learning the initial landmark set by analyzing the input data sets, semantics, and user preferences. We also plan to introduce learning for the tuning of our `Adaptive` scheme parameters, which will further improve its sampling component and overall utility performance.

Bibliography

[ABCP13] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.

[ABN+08] Osman Abul, Francesco Bonchi, Mirco Nanni, et al. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, volume 8, pages 376–385, 2008.

[acx21] Acxiom, Accessed on October 11, 2021. URL: `https://acxiom.com`.

[ADC18] Raed Al-Dhubhani and Jonathan M Cazalas. An adaptive geo-indistinguishability mechanism for continuous lbs queries. *Wireless Networks*, 24(8):3221–3239, 2018.

[AMX+20] Nadeem Ahmed, Regio A Michelin, Wanli Xue, Sushmita Ruj, Robert Malaney, Salil S Kanhere, Aruna Seneviratne, Wen Hu, Helge Janicke, and Sanjay K Jha. A survey of covid-19 contact tracing apps. *IEEE access*, 8:134577–134601, 2020.

[Ans95] Luc Anselin. Local indicators of spatial association—lisa. *Geographical analysis*, 27(2):93–115, 1995.

[BA05] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.

[BBDS13] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 87–96. ACM, 2013.

[BCHL09] Josh Benaloh, Melissa Chase, Eric Horvitz, and Kristin Lauter. Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 103–114. ACM, 2009.

[BEM+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459. ACM, 2017.

[BFM+13] Jean Bolot, Nadia Fawaz, S Muthukrishnan, Aleksandar Nikolov, and Nina Taft. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory*, pages 284–295. ACM, 2013.

[Blo70] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[BP66] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[CAC12] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 638–649. ACM, 2012.

[CEP+17] Kostantinos Chatzikokolakis, Ehab ElSalamouny, Catuscia Palamidessi, Pazii Anna, et al. Methods for location privacy: A comparative overview. *Foundations and Trends® in Privacy and Security*, 1(4):199–257, 2017.

[CFD11] Rui Chen, Benjamin Fung, and Bipin C Desai. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020*, 2011.

[CFYD14] Rui Chen, Benjamin C Fung, Philip S Yu, and Bipin C Desai. Correlated network data publication via differential privacy. *The VLDB Journal-The International Journal on Very Large Data Bases*, 23(4):653–676, 2014.

[CM11] Chi-Yin Chow and Mohamed F Mokbel. Trajectory privacy in location-based services and data publication. *ACM Sigkdd Explorations Newsletter*, 13(1):19–29, 2011.

[CMHM17] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. Pegasus: Data-adaptive differentially private stream processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1375–1388. ACM, 2017.

[CPS15] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Geo-indistinguishability: A principled approach to location privacy. In *International Conference on Distributed Computing and Internet Technology*, pages 49–72. Springer, 2015.

[CRKH11] Delphine Christin, Andreas Reinhardt, Salil S Kanhere, and Matthias Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of systems and software*, 84(11):1928–1946, 2011.

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):26, 2011.

[CWL+14] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems*, 25(1):222–233, 2014.

[CY15] Yang Cao and Masatoshi Yoshikawa. Differentially private real-time data release over infinite trajectory streams. In *Mobile Data Management (MDM), 2015 16th IEEE International Conference on*, volume 2, pages 68–73. IEEE, 2015.

[CYXX17] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying differential privacy under temporal correlations. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*, pages 821–832. IEEE, 2017.

[CYXX18] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying differential privacy in continuous data release under temporal correlations. *IEEE Transactions on Knowledge and Data Engineering*, 31(7):1281–1295, 2018.

[DMHVB13] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DNP+10] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *ICS*, pages 66–80, 2010.

[DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724. ACM, 2010.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Dwo08] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[eco16] The world's most valuable resource is no longer oil, but data. `https://economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data`, 2016. Accessed on October 11, 2021.

[EF15] Murat A Erdogdu and Nadia Fawaz. Privacy-utility trade-off under continual observation. In *ISIT*, pages 1801–1805, 2015.

[EK03] Ken TD Eames and Matt J Keeling. Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1533):2565–2571, 2003.

[EL18] Fatima Zahra Errounda and Yan Liu. Continuous location statistics sharing algorithm with local differential privacy. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5147–5152. IEEE, 2018.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[ESC15] Vasilis Efthymiou, Kostas Stefanidis, and Vassilis Christophides. Big data entity resolution: From highly to somehow similar entity descriptions in the web. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 401–410. IEEE, 2015.

[exp21] Experian, Accessed on October 11, 2021. URL: `https://experian.com`.

[fac21] Facebook, Accessed on October 11, 2021.

[Far20] Farhad Farokhi. Temporally discounted differential privacy for evolving datasets on an infinite horizon. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–8. IEEE, 2020.

[FB15] Lorenzo Franceschi-Bicchierai. Redditor cracks anonymous data trove to pinpoint muslim cab drivers. `https://mashable.com/2015/01/28/redditor-muslim-cab-drivers`, 2015. Accessed on October 11, 2021.

[FKZ+19] Marco Fiore, Panagiota Katsikouli, Elli Zavou, Mathieu Cunche, Françoise Fessant, Dominique Le Hello, Ulrich Matchi Aivodji, Baptiste Olivier, Tony Quertier, and Razvan Stanica. Privacy of trajectory micro-data: a survey. *arXiv preprint arXiv:1903.12211*, 2019.

[fou21] Foursquare, Accessed on October 11, 2021. URL: `https://foursquare.com`.

[FWFP08] Benjamin Fung, Ke Wang, Ada Wai-Chee Fu, and Jian Pei. Anonymity for continuous data publishing. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 264–275. ACM, 2008.

[FX14] Liyue Fan and Li Xiong. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2094–2106, 2014.

[FXS13] Liyue Fan, Li Xiong, and Vaidy Sunderam. Differentially private multi-dimensional time series release for traffic monitoring. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 33–48. Springer, 2013.

[Gag17] Paul A Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[GDSB09] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. Preventing velocity-based linkage attacks in location-aware applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 246–255. ACM, 2009.

[GKdPC10] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41, 2010.

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam 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.

[GLTY18] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. Differentially private and utility preserving publication of trajectory data. *IEEE Transactions on Mobile Computing*, 2018.

[gma21] Google maps, Accessed on October 11, 2021. URL: `https://google.com/maps`.

[GNG12] Michaela Götz, Suman Nath, and Johannes Gehrke. Maskit: Privately releasing user context streams for personalized mobile applications. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 289–300. ACM, 2012.

[Gol98] Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78, 1998.

[Grü07] Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

[Gut13] Allan Gut. *Probability: a graduate course*, volume 75. Springer Science & Business Media, 2013.

[GWO00] George D Gaskell, Daniel B Wright, and Colm A O'Muircheartaigh. Telescoping of landmark events: implications for survey research. *The Public Opinion Quarterly*, 64(1):77–89, 2000.

[HBN11] Yeye He, Siddharth Barman, and Jeffrey Naughton. Preventing equivalence attacks in updated, anonymized data. 2011.

[HC01] Shyamanta M Hazarika and Anthony G Cohn. Qualitative spatio-temporal continuity. In *International Conference on Spatial Information Theory*, pages 92–107. Springer, 2001.

[HCM+15] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M Procopiuc, and Divesh Srivastava. Dpt: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment*, 8(11):1154–1165, 2015.

[HGZ15] Jingyu Hua, Yue Gao, and Sheng Zhong. Differentially private publication of general time-serial trajectory data. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 549–557. IEEE, 2015.

[JGK16] Priyank Jain, Manasi Gyanchandani, and Nilay Khare. Big data privacy: a technological perspective and review. *Journal of Big Data*, 3(1):25, 2016.

[JLE14] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[JNS18] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.

[JSB+13] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. Publishing trajectories with differential privacy guarantees. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, page 12. ACM, 2013.

[Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[KCTK16] Manos Katsomallos, Vassilis Christophides, Katerina Tzompanaki, and Dimitris Kotzinos. Measuring privacy leakage under continual publication of crowdsensing data, 2016. São Paulo School of Advanced Science on Smart Cities.

[KHLF10] Himanshu Khurana, Mark Hadley, Ning Lu, and Deborah A Frincke. Smart-grid security issues. *IEEE Security & Privacy*, 8(1):81–85, 2010.

[Kif09] Daniel Kifer. Attacks on privacy and definetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 127–138. ACM, 2009.

[Kin83] John Leslie King. Centralized versus decentralized computing: organizational considerations and management options. *ACM Computing Surveys (CSUR)*, 15(4):319–349, 1983.

[KL10] Seny Kamara and Kristin Lauter. Cryptographic cloud storage. In *International Conference on Financial Cryptography and Data Security*, pages 136–149. Springer, 2010.

[KLPT17] Manos Katsomallos, Spyros Lalis, Thanasis Papaioannou, and George Theodorakopoulos. An open framework for flexible plug-in privacy mechanisms in crowdsensing applications. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 237–242. IEEE, 2017.

[KM11] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204. ACM, 2011.

[KM14] Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):3, 2014.

[KP13] Georgios Kellaris and Stavros Papadopoulos. Practical differential privacy via grouping and smoothing. In *Proceedings of the VLDB Endowment*, volume 6, pages 301–312. VLDB Endowment, 2013.

[KPXP14] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment*, 7(12):1155–1166, 2014.

[KTK16] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. Data quality and user privacy in big geodata: How does the one affect the other?, 2016. 11th International Workshop on Information Search, Integration, and Personalization.

[KTK17] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. Data quality issues in big geodata: how does this affect privacy?, 2017. DaQuaTa International Workshop.

[KTK19] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. Privacy, space and time: a survey on privacy-preserving continuous data publishing. *Journal of Spatial Information Science*, 2019(19):57–103, 2019.

[KTK22] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. Landmark privacy: Configurable differential privacy protection for time series. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, 2022. To appear.

[LBS+16] Jiuyong Li, Muzammil M Baig, AHM Sarowar Sattar, Xiaofeng Ding, Jixue Liu, and Millist W Vincent. A hybrid approach to prevent composition attacks for independent data releases. *Information Sciences*, 367:324–336, 2016.

[LC11] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[LCM16] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Dependence makes you vulnerable: Differential privacy under dependent tuples. In *NDSS*, volume 16, pages 21–24, 2016.

[Leg93] Pierre Legendre. Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6):1659–1673, 1993.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[LMP01] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[LSP+07] Feifei Li, Jimeng Sun, Spiros Papadimitriou, George A Mihaila, and Ioana Stanoi. Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 686–695. IEEE, 2007.

[Lyo14] David Lyon. Surveillance, snowden, and big data: Capacities, consequences, critique. *Big Data & Society*, 1(2):2053951714541861, 2014.

[LZX+08] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–10, 2008.

[LZZX17] Meng Li, Liehuang Zhu, Zijian Zhang, and Rixin Xu. Achieving differential privacy of trajectory data publishing in participatory sensing. *Information Sciences*, 400:1–13, 2017.

[Mak18] Stephen Makonin. Hue: The hourly usage of energy dataset for buildings in british columbia. Technical report, 2018.

[McS09] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[MGKV06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[MGM16] Holger Kisker Mike Gualtieri, Rowan Curran and Emily Miller. Perishable insights – stop wasting money on unactionable analytics, 2016.

[MI15] Kourosh Meshgi and Shin Ishii. Expanding histogram of colors with gridding to improve tracking accuracy. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 475–479. IEEE, 2015.

[Mor50] Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

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

[MZL+19] Zhuo Ma, Tian Zhang, Ximeng Liu, Xinghua Li, and Kui Ren. Real-time privacy-preserving data release over vehicle trajectory. *IEEE transactions on vehicular technology*, 68(8):8091–8102, 2019.

[MZZ+17] Qiang Ma, Shanfeng Zhang, Tong Zhu, Kebin Liu, Lan Zhang, Wenbo He, and Yunhao Liu. Plp: Protecting location privacy against

correlation analyze attack in crowdsensing. *IEEE transactions on mobile computing*, 16(9):2588–2598, 2017.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse data sets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[NYER19] Carolina Naim, Fangwei Ye, and Salim El Rouayheb. On-off privacy with correlated requests. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 817–821. IEEE, 2019.

[OQL+18] Lu Ou, Zheng Qin, Shaolin Liao, Hui Yin, and Xiaohua Jia. An optimal pufferfish privacy mechanism for temporally correlated trajectories. *IEEE Access*, 6:37150–37165, 2018.

[osm21] Openstreetmap, Accessed on October 11, 2021. URL: `https://openstreetmap.org`.

[PBMB18] Vincent Primault, Antoine Boutet, Sonia Ben Mokhtar, and Lionel Brunie. The long road to computational location privacy: A survey. *IEEE Communications Surveys & Tutorials*, 2018.

[PMLB15] Vincent Primault, Sonia Ben Mokhtar, Cédric Lauradoux, and Lionel Brunie. Time distortion anonymization for the publication of mobility data with high utility. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 539–546. IEEE, 2015.

[PP18] Kun Il Park and Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018.

[QBB+17] Do Le Quoc, Martin Beck, Pramod Bhatotia, Ruichuan Chen, Christof Fetzer, and Thorsten Strufe. Privapprox: privacy-preserving stream analytics. In *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference*, pages 659–672. USENIX Association, 2017.

[Qui14] J Ross Quinlan. *Programs for machine learning*. Elsevier, 2014.

[rin21] Ring, Accessed on October 11, 2021. URL: `https://ring.com`.

[Rus18] Jon Russell. Fitness app Strava exposes the location of military bases. `https://techcrunch.com/2018/01/28/strava-exposes-military-bases`, 2018. Accessed on October 11, 2021.

[RW00] L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press, 2000.

[Sat17] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[Sca81] Theodore Scaltsas. Identity, origin and spatiotemporal continuity. *Philosophy*, 56(217):395–402, 1981.

[SCDF16] Jordi Soria-Comas and Josep Domingo-Ferrer. Big data privacy: challenges to privacy principles and models. *Data Science and Engineering*, 1(1):21–28, 2016.

[SCOL+04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004.

[Sha01] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[Sko05] Valeriy Skorokhod. *Basic principles and applications of probability theory*. Springer Science & Business Media, 2005.

[SNE17] Ms MS Simi, Mrs K Sankara Nayaki, and M Sudheep Elayidom. An extensive study on data anonymization algorithms based on k-anonymity. In *IOP Conference Series: Materials Science and Engineering*, volume 225, page 012279. IOP Publishing, 2017.

[SSLL19] Piotr Sapiezynski, Arkadiusz Stopczynski, David Dreyer Lassen, and Sune Lehmann. Interaction data from the copenhagen networks study. *Scientific Data*, 6(1):1–10, 2019.

[ST15] Erez Shmueli and Tamir Tassa. Privacy by diversity in sequential releases of databases. *Information Sciences*, 298:344–372, 2015.

[Sti89] Stephen M Stigler. Francis galton's account of the invention of correlation. *Statistical Science*, pages 73–79, 1989.

[STW+12] Erez Shmueli, Tamir Tassa, Raz Wasserstein, Bracha Shapira, and Lior Rokach. Limiting disclosure of sensitive data in sequential releases of databases. *Information Sciences*, 191:98–127, 2012.

[SWC17] Shuang Song, Yizhen Wang, and Kamalika Chaudhuri. Pufferfish privacy mechanisms for correlated data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1291–1306. ACM, 2017.

[Swe02a] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.

[Swe02b] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[Tan16] Colin Tankard. What the gdpr means for businesses. *Network Security*, 2016(6):5–8, 2016.

[Tob70] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[tou21] Tousanticovid, Accessed on October 11, 2021. URL: `https://bonjour.tousanticovid.gouv.fr`.

[tra21] Transunion, Accessed on October 11, 2021. URL: `https://transunion.com`.

[twi21] Twitter, Accessed on October 11, 2021. URL: `https://twitter.com`.

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

[waz21] Waze, Accessed on October 11, 2021. URL: `https://waze.com`.

[WBLJ17] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 729–745, 2017.

[WCFY10] K Wang, R Chen, BC Fung, and PS Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, 2010.

[Wei06] William WS Wei. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. 2006.

[WF06] Ke Wang and Benjamin Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 414–423. ACM, 2006.

[Wig67] David Wiggins. *Identity and spatio-temporal continuity*. Blackwell Oxford, 1967.

[wik21] Wikipedia, Accessed on October 11, 2021. URL: `https://wikipedia.com`.

[WLZL09] Jian Wang, Yongcheng Luo, Yan Zhao, and Jiajin Le. A survey on privacy preserving data mining. In *Database Technology and Applications, 2009 First International Workshop on*, pages 111–114. IEEE, 2009.

[WSN18] Shuo Wang, Richard Sinnott, and Surya Nepal. Privacy-protected statistics publication over social media user trajectory streams. *Future Generation Computer Systems*, 87:792–802, 2018.

[WX17] Hao Wang and Zhengquan Xu. Cts-dp: publishing correlated time-series data via differential privacy. *Knowledge-Based Systems*, 122:167–179, 2017.

[WXJ+21] Hao Wang, Zhengquan Xu, Shan Jia, Ying Xia, and Xu Zhang. Why current differential privacy schemes are inapplicable for correlated data publishing? *World Wide Web*, 24:1–23, 2021.

[WZL+16] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[xss21] @xssfopes: "can anyone spot the issue with the algo? red is original data point, 400 "anonymized" data points calculated", Accessed on October 11, 2021. URL: `https://twitter.com/xssfox/status/1251116087116042241`.

[XT07] Xiaokui Xiao and Yufei Tao. M-invariance: towards privacy preserving re-publication of dynamic data sets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 689–700. ACM, 2007.

[XX15] Yonghui Xiao and Li Xiong. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1298–1309. ACM, 2015.

[XXZC17] Yonghui Xiao, Li Xiong, Si Zhang, and Yang Cao. Loclok: location cloaking with differential privacy via hidden markov model. *Proceedings of the VLDB Endowment*, 10(12):1901–1904, 2017.

[YLX+17] Ayong Ye, Yacheng Li, Li Xu, Qing Li, and Hui Lin. A trajectory privacy-preserving algorithm based on road networks in continuous location-based services. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 510–516. IEEE, 2017.

[YNER19] Fangwei Ye, Carolina Naim, and Salim El Rouayheb. Preserving on-off privacy for past and future requests. In *2019 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2019.

[YNER21] Fangwei Ye, Carolina Naim, and Salim El Rouayheb. On-off privacy against correlation over time. *IEEE Transactions on Information Forensics and Security*, 16:2104–2117, 2021.

[YNR20] Fangwei Ye, Carolina Naim, and Salim El Rouayheb. On-off privacy in the presence of correlation. *arXiv preprint arXiv:2004.04186*, 2020.

[YZZ+10] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108, 2010.

[ZCP+17] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):25, 2017.

[Zhe15] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[ZHP+09] Bin Zhou, Yi Han, Jian Pei, Bin Jiang, Yufei Tao, and Yan Jia. Continuous privacy preserving publishing of data streams. In *Proceedings of the 12th International Conference on Extending Database*

Technology: Advances in Database Technology, pages 648–659. ACM, 2009.

[ZPL08] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 10(2):12–22, 2008.

[ZZP17] Jun Zhao, Junshan Zhang, and H Vincent Poor. Dependent differential privacy for correlated data. In *Globecom Workshops (GC Wkshps), 2017 IEEE*, pages 1–7. IEEE, 2017.