Privacy-Preserving Release of Spatio-temporal Density

Gergely Acs, Gergely Biczók, and Claude Castelluccia

Abstract In today's digital society, increasing amounts of contextually rich spatio-temporal information are collected and used, e.g., for knowledge-based decision making, research purposes, optimizing operational phases of city management, planning infrastructure networks, or developing timetables for public transportation with an increasingly autonomous vehicle fleet. At the same time, however, publishing or sharing spatio-temporal data, even in aggregated form, is not always viable owing to the danger of violating individuals' privacy, along with the related legal and ethical repercussions. In this chapter, we review some fundamental approaches for anonymizing and releasing spatio-temporal density, i.e., the number of individuals visiting a given set of locations as a function of time. These approaches follow different privacy models providing different privacy guarantees as well as accuracy of the released anonymized data. We demonstrate some sanitization (anonymization) techniques with provable privacy guarantees by releasing the spatio-temporal density of Paris, in France. We conclude that, in order to achieve meaningful accuracy, the sanitization process has to be carefully customized to the application and public characteristics of the spatio-temporal data.

Gergely Acs
CrySyS Lab, Department of Networked Systems and Services, Budapest University of Technology and Economics (BME-HIT), Magyar tudosok korutja 2., 1117 Budapest, Hungary, e-mail: acs@crysys.hu

Gergely Biczók
CrySyS Lab, Department of Networked Systems and Services, Budapest University of Technology and Economics (BME-HIT), Magyar tudosok korutja 2., 1117 Budapest, Hungary, e-mail: biczok@crysys.hu

Claude Castelluccia
INRIA, 655 Avenue de l'Europe, 38330 Montbonnot-Saint-Martin, France, e-mail: claude.castelluccia@inria.fr

1 Introduction

Spatio-temporal, geo-referenced datasets are growing rapidly nowadays. With billions of location-aware devices in use worldwide, the large scale collection of space-time trajectories of people produces gigantic mobility datasets. Such datasets are invaluable for traffic and sustainable mobility management, or studying accessibility to services. Even more, they can help understand complex processes, such as the spread of viruses or how people exchange information, interact, and develop social interactions. While the benefits provided by these datasets are indisputable, their publishing or sharing is not always viable owing to the danger of violating individuals' privacy, along with the related legal and ethical repercussions. This problem is socially relevant: companies and researchers are reluctant to publish any mobility data by fear of being held responsible for potential privacy breaches. This limits our ability to analyze such large datasets to derive information that could benefit the general public.

Unsurprisingly, personal mobility data reveals tremendous sensitive information about individuals' behavioural patterns such as health life or religious/political beliefs. Somewhat more surprisingly, such mobility data is also unique to individuals even in a relatively large population containing millions of users. For instance, only four spatio-temporal positions are enough to uniquely identify a user 95% of the times in a dataset of one and a half million users [13], even if the dataset is *pseudonymized*, i.e., identitifiers such as personal names, phone numbers, home address are suppressed. Moreover, the top 2 mostly visited locations of an individual is still unique with a probability of 10-50% [63] among millions of users. Notice that the most visited locations, such as home and working places, are easy to learn today from different social media where people often publicly reveal this seemingly harmless personal information. Therefore, publishing mobility datasets would put at risk our own privacy; if someone knows where we live and work could potentially find our record and learn all of our potentially sensitive location visits. Moreover, due to the large uniqueness of records, these datasets are regarded as personal information under several laws and regulations internationally, such as overall in the European Union. Therefore, their release prompt not only serious privacy concerns but also possible monetary penalties [18].

1.1 Privacy implications of aggregate location data

One might argue that publishing aggregate information, such as the number of individuals at a given location, is enough to reconstruct aggregate mobility patterns, and has no privacy implications. Indeed, aggregated information is usually related to large groups of individuals and is seemingly safe to disclose. However, this reasoning is flawed as shown next. First, an attack is described that can reconstruct even entire individual trajectories from aggregate location data, if aggregates are periodically and sufficiently frequently published (e.g., in every half an hour). We also

illustrate the potential privacy threats of irregularly published aggregate location data, for example, when a querier (or the adversary) specifies the spatio-temporal points whose visits are then aggregated and released.

Consequently, aggregation *per se* do not necessarily prevent privacy breaches, and we need additional countermeasures to guarantee privacy for individuals even in a dataset of aggregate mobility data such as spatio-temporal densities.

1.1.1 Reconstruction from periodically published aggregate data

The attack described in [61] successfully reconstructed more than 70% of 100 000 trajectories merely from the total number of visits at 8000 locations, which were published every half an hour over a whole week in a large city. The attack exploits three fundamental properties of location trajectories:

Predictability: The current location of an individual can be accurately predicated from his previous location because consecutively visited locations are usually geographically close. This implies that trajectories can be well-separated in space; if two trajectories are far away in time t then they remain so in time $(t+1)$ assuming that t and $t+1$ are not too distant in time.

Regularity: Most people visit very similar (or the same) locations every day. Indeed, human mobility is governed by daily routines and hence periodic. For example, people go to work/school and return home at almost the same time every day.

Uniqueness: Every person visits quite different locations than any other person even in a very large population, which has already been demonstrated by several studies. For example, any four locations of an individual trajectory are unique to that trajectory with a probability of more than 95% for one and a half million individuals [13].

The attack has three main phases. In the first phase, it reconstructs every trajectory within every single day by exploiting the predictability of trajectories. This is performed by finding an optimal match of locations between consecutive time slots, where geographically close locations are more likely to be matched. After the first phase, we have the daily fragments of every trajectory, but we do not know which fragments belong to the same trajectory. Hence, in the second phase, complete trajectories are reconstructed by identifying their daily fragments. This is feasible due to the regularity and uniqueness properties of trajectories, i.e. every trajectory has similar daily fragments which are also quite different from the fragments of other trajectories. Similarity of fragments can be measured by the frequency of visits per location within a fragment. Finally, in the last phase, re-identification of individuals are carried out by using the uniqueness property again; a few locations of any individual known from external sources (e.g., social media) will single out the individual's trajectory [13]. As individual trajectories are regarded as personal data in several regulations internationally, the feasibility of this attack demonstrates that aggregate location data can also be regarded as personal data.

1.1.2 Reconstruction from irregularly published aggregate data

Another approach of releasing spatio-temporal density is to answer some counting queries executed on the location trajectories. The querier is interested in the number of people whose trajectories satisfy a specified condition (e.g., the number of trajectories which contain a certain hospital). Queries can be filtered instantly by an auditor, e.g. all queries which have too small support, say less than k (i.e., only k trajectories satisfy the condition), are simply refused to answer. However, this approach is not enough to prevent privacy breaches; if the support of two queries are both greater than k, their difference can still be 1. For instance, the first query may ask for the number of people who visited a hospital, and the second query for the number of people who visited the same hospital except locations L_1 and L_2. If the querier knows that L_1 and L_2 are unique to John then it learns whether John visited the hospital.

Defenses against such *differencing attacks* are not straightforward. For example, verifying whether the answers of two or more queries disclose any location visit can be computationally infeasible; if the query language is sufficiently complex there is no efficient algorithm to decide whether two queries constitute a differencing attack [30]. In Section 3.1, we show more principled techniques to recover individual location visits from the answers of a given query set.

1.2 Applications of spatio-temporal density

Spatio-temporal density data, albeit aggregated in nature, can enable a wide variety of optimization use cases by providing a form of location awareness, especially in the context of the Smart City concept [46]. Depending on both its spatial and temporal granularity, such data can be useful for optimizing the (i) design and/or (ii) operational phases of city management with regard to e.g., public transportation, local businesses or emergency preparedness. Obviously, spatial resolution determines the scale of such optimization, e.g., whether we can tell a prospective business owner to open her new cafe in a specific district or a specific street. On the other hand, it is the temporal granularity of density data that separates the application scenarios in terms of design and operational use cases.

In case of low temporal granularity (i.e., not more than a few data points per area per day), city officials can use the data for optimizing design tasks such as:

- planning infrastructure networks, such as new roads, railways or communication networks;
- advising on the location of new businesses such as retail, entertainment and food;
- developing timetables for public transportation;
- deploying hubs for urban logistics systems such as post, vehicle depos (e.g., for an urban bike rental system), electric vehicle chargers and even city maintenance personnel;

In case of high temporal granularity (i.e., several data points per area per hour) [33], spatio-temporal density data might enable on-the-fly operational optimization in the manner of:

- reacting to and forecasting traffic-related phenomena including traffic anomaly detection and re-routing;
- implementing adaptive public transportation timetables also with an increasingly autonomous vehicle fleet [52];
- scheduling maintenance work adaptively causing the least amount of disturbance to inhabitants;
- promoting energy efficiency by switching off unneeded electric equipment on-demand (cell towers, escalators, street lighting);
- location-aware emergency preparedness protocols in case of natural disasters or terrorist attacks [7].

These lists of application scenarios are not comprehensive. Interestingly, such an aggregated view on human mobility enables a large set of practical applications.

2 Privacy models

Privacy has a multitude of definitions, and thus different privacy models have been proposed. In terms of privacy guarantee, we distinguish between syntactic and semantic privacy models. Syntactic models focus on syntactic requirements of the anonymized data (e.g., each record should appear at least k times in the anonymized dataset) without any guarantee on what sensitive information the adversary can exactly learn about individuals. As opposed to this, semantic models[1] are concerned with the private information that can be inferred about individuals using the anonymized data as well as perhaps some prior (or background) knowledge about them. The commonality of all privacy models is the inherent trade-off between privacy and utility: guaranteeing any meaningful privacy requires the distortion of the original dataset which yields imprecise, coarse-grained knowledge even about the population as a whole. There is no free lunch: perfect privacy with maximally accurate anonymized data is impossible. Each model has different privacy guarantees and hence provide different accuracy of the (same) data.

[1] In our context, semantic privacy is not analogous to semantic security used in cryptography, where ciphertexts must not leak any information about plaintexts. Anonymized data ("ciphertext") should allow partial information leakage about the original data ("plaintext"), otherwise any data release would be meaningless. Such partial leakage should include the release of useful population (and not individual specific) characteristics.

2.1 Syntactic privacy models

One of the most influential privacy model is k-anonymity, which was first introduced in computer science by [53], albeit the same notion had already existed before in statistical literature. In general, for location data, k-anonymity guarantees that any record is indistinguishable with respect to spatial and temporal information from at least $k-1$ other records. Hence, an adversary who knows some attributes of an individual (such as few visited places) may not be able decide which record belongs to this person. Now, let us define k-anonymity more formally.

Definition 1 (k-anonymity [53]). Let $\mathbb{P} = \{P_1, \ldots, P_{|\mathbb{P}|}\}$ be a set of public attributes, and $\mathbb{S} = \{S_1, \ldots, S_{|\mathbb{S}|}\}$ be a set of sensitive attributes. A relational table $R(\mathbb{P}, \mathbb{S})$ satisfies k-anonymity iff, for each record in r in R, there are at least $k-1$ other records in R which have the same public attribute values as r.

k-anonymity requires (syntactic) indistinguishability of every record in the dataset from at least $k-1$ other records with respect to their public attributes. Originally, public attributes included all (quasi)-identifiers of an individual (such as sex, ZIP code, birth date) which are easily learnable by an adversary, while the sensitive attribute value (e.g., salary, medical diagnosis, etc.) of any individual should not be disclosed. Importantly, the values of public attributes are likely to be unique to a person in a population [23], and hence can be used to link multiple records of the same individual across different datasets, if these datasets share common public attributes. In the context of location data, where a spatio-temporal point (L,t) corresponds to a binary attribute whose value is 1 if the individual visited location L at time t and 0 otherwise, such distinction of public and sensitive attributes is usually pointless. Indeed, the same location can be insensitive to one person while sensitive to another one (e.g., a hospital may be an insensitive place for a doctor, who works there, and sensitive for a patient). Therefore, in a location dataset, k-anonymity should require that each record (trajectory) must be completely identical to at least $k-1$ other trajectories in the same dataset. Syntactically indistinguishable trajectories/records form a single anonymity group.

k-anonymity can be achieved by generalizing and/or suppressing the location visits of individuals in the anonymized dataset. Generalization can be performed by either forming clusters of similar trajectories, where each cluster has at least k trajectories, or by replacing the location and/or time information of trajectories with a less specific, but semantically consistent, one. For example, cities are represented by their county, whereas minutes or hours are represented by the time of day (morning/afternoon/evening/night).

A relaxation of k-anonymity, called k^m-anonymity, was first proposed in [54]. This model imposes an explicit constraint on the background knowledge of the adversary, and requires k-anonymity with respect to this specific knowledge. For example, if the adversary can learn at most m location visits of an individual, then, for any set of m location visits, there must be at least 0 or k records in the anonymized dataset which contain this particular set of visits. Formally:

Definition 2 (k^m-anonymity [54]). Given a dataset D where each record is subset of items from a universe $\mathscr{U}$. D is k^m-anonymous iff for any m items from $\mathscr{U}$ there are 0 or at least k records which contain these items.

In our context, universe $\mathscr{U}$ represents all spatio-temporal points, and an individual's record has an item from U if the corresponding spatio-temporal is visited by the individual.

No.	Locations
1	{LA}
2	{LA, Seattle}
3	{NYC, Boston}
4	{NYC, Boston}
5	{LA, Seattle, NYC}
6	{LA, Seattle, NYC}
7	{LA, Seattle, NYC, Boston}

(a) Original

No.	Locations
1	{West US}
2	{West US}
3	{NYC, Boston}
4	{NYC, Boston}
5	{LA, Seattle, West US}
6	{LA, Seattle, West US}
7	{LA, Seattle, West US}

(b) 2-anonymous

No.	Locations
1	{LA}
2	{LA, Seattle}
3	{West US}
4	{West US}
5	{LA, Seattle, West US}
6	{LA, Seattle, West US}
7	{LA, Seattle, West US}

(c) 2^2-anonymous

Table 1: Examples for k- and k^m-anonymity, where each row represents a record, public and sensitive attributes are not distinguished, and temporal information is omitted for simplicity. 2^2-anonymity requires fewer generalizations and hence provides more accurate data at the cost of privacy.

If m equals the maximum number of location visits per record, then k^m-anonymity boils down to standard k-anonymity. However, the rationale behind k^m-anonymity is that the adversary is usually incapable of learning more than a few locations visits per individual (e.g., most people publicly reveal only their home and working places on social media, in which case $m = 2$ if temporal data is disregarded). Clearly, requiring indistinguishability with respect to only m instead of all location visits of an individual requires less generalization and/or suppression thereby providing more accurate anonymized data. This is also illustrated in Table 1.

We must note that many more different syntactic privacy models (e.g., ℓ-diversity [39], t-closeness [37], (L,K,C)-privacy [42], etc.) have been proposed to mitigate the deficiencies of k-anonymity. We refer the interested reader to [21] and [56] for more details on privacy models and their usage. In this chapter, we only consider syntactic anonymization schemes which rely on k- or k^m-anonymity.

2.2 Semantic privacy models

Most syntactic privacy models, such as k-anonymity, aim to mitigate only identity disclosure, when the adversary re-identifies a record in the dataset (i.e., infer the exact identity of the record owner). Although re-identification is clearly undesirable and explicitly addressed by most legal regulations worldwide, it is not a necessary

condition of privacy violations. That is, locating the anonymity group of a person (e.g., using his home and working places), the group itself can still leak a person's visited places no matter how large the group is. For instance, each of the k trajectory may contain the same sensitive place, which means that the person also passed this place. The real culprit is the lack of uncertainty about the individuals' presence in the *anonymized* dataset; even a knowledgeable adversary, who may know that a person's record is part of the original dataset, should not be able learn if this record was indeed used to generate the anonymized data. Another common pitfall of syntactic privacy models is the lack of *composability*; the privacy of independent releases of the same or correlated datasets should not collapse but rather "degrade gracefully". However, this does not hold for k-anonymity: the composition of k-anonym datasets, where k can be arbitrarily large, can only be 1-anonym (i.e., the anonymity guarantee completely collapses), which is also demonstrated in [22]. Composability is a natural requirement of any privacy model in the era of Big Data where many different pieces of personal data get anonymized and published about people by many different stakeholders independently. These different pieces may be gathered and combined by a knowledgeable adversary in order to breach individuals' privacy. Next, we present a model which addresses these concerns.

Intuitively, differential privacy [15] requires that the outcome of any computation be insensitive to the change of any single record inside and outside the dataset. It allows a party to privately release a dataset: with perturbation mechanisms, a function of an input dataset is modified, prior to its release, so that any information which can discriminate a record from the rest of the dataset is bounded [16].

Definition 3 (Differential Privacy [16]). A privacy mechanism $\mathscr{A}$ guarantees (ε, δ)-differential privacy if for any database D and D', differing on at most one record, and for any possible output $S \subseteq Range(\mathscr{A})$,

$$Pr[\mathscr{A}(D) \in S] \leq e^{\varepsilon} \times Pr[\mathscr{A}(D') \in S] + \delta$$

or, equivalently, $\Pr_{O \sim \mathscr{A}(D)} \left[\log \left(\frac{\Pr[\mathscr{A}(D)=O]}{\Pr[\mathscr{A}(D')=O]} \right) > \varepsilon \right] \leq \delta$.

Here, ε is typically a modest value (i.e., less than 1), and δ is a negligible function of the number of records in D (i.e., less then $1/|D|$) [16].

We highlight two consequences of the above definition which are often overlooked or misinterpreted. First, differential privacy guarantees plausible deniability to every individual *inside as well as outside* of the dataset, as an adversary, provided with the output of $\mathscr{A}$, can draw almost the same conclusions about any individual no matter if this individual is included in the input of $\mathscr{A}$ or not [16]. Specifically, Definition 3 guarantees that every output of algorithm $\mathscr{A}$ is almost equally likely (up to ε) on datasets differing in a single record except with probability at most δ. This implies that *every possible* binary inference (i.e., predicate) has almost the same probability to be true (false) on neighboring datasets [15]. For example, if an adversary can infer from $\mathscr{A}(D)$ that an individual, say John, visited a hospital with probability 0.95, where D excludes John's record, then the same adversary infers the same from $\mathscr{A}(D')$ with probability $\approx e^{\pm\varepsilon} \times 0.95 + \delta$, where $D' = D \cup \{\text{John's record}\}$.

This holds for *any* adversary and inference irrespective of the applied inference algorithm and prior (background) knowledge[2]. That is, the privacy measure ε and δ are "agnostic" to the adversarial background knowledge and inference algorithm.

Second, Definition 3 *does not* provide any guarantee about the (in)accuracy of any inference. There can be inferences (adversaries) which may predict the hospital visit of John quite accurately, e.g., by noticing that all records, which are very similar to John's record (such as the records having the same age and profession as John), also visited a hospital [11], while other inferences may do a bad job of prediction as they cannot reliably sort out the records being similar (correlated) to John's record. Definition 3 guarantees that the accuracy of *any* inferences, no matter how sensitive are, remain unchanged (up to ε and δ) if John's own record is included in the anonymized data. In other words, differential privacy allows to learn larger statistical trends in the dataset, even if these trends reveal perhaps sensitive information about each individual, and protects secrets about individuals which can only be revealed with their participation in the dataset[3]. Learning such trends (i.e., inferences which are generalizable to a larger population in interest) is the ultimate goal of any data release in general.

Therefore, the advantage of differential privacy, compared to the many other models proposed in the literature, is two-fold. First, it provides a formal and measurable privacy guarantee regardless what other background information or sophisticated inference technique the adversary uses even in the future. Second, following from Definition 3, it is closed with respect to sequential and parallel composition, i.e., the result of the sequential or parallel combination of two differential private algorithms is also differential private.

Theorem 1 ([40]). *If each of $\mathscr{A}_1, \ldots, \mathscr{A}_k$ is (ε, δ)-differential private, then their k-fold adaptive composition*[4] *is $(k\varepsilon, k\delta)$-differential private.*

Composition property has particular importance in practice, since it does not only simplify the design of anonymization (sanitization) solutions, but also allows to measure differential privacy when a given dataset, or a set of correlated datasets, is anonymized (and released) several times, possibly by different entities.

There are a few ways to achieve DP and all of them are based on the randomization of a computation whose result ought to be released. Most of these techniques are composed of adding noise to the true output with zero mean and variance calibrated to desired privacy guarantee which is measured by ε and δ. A fundamental concept of these techniques is the *global sensitivity* of the computation (function) [16] whose result should be released:

---

[2] The inference algorithm and background knowledge influences only the probability of the conclusion, which is 0.95 in the current example

[3] These secrets are the *private* information which discriminate the individual from the rest of the dataset and should be protected

[4] Adaptive composition means that the output of $\mathscr{A}_{i-1}$ is used as an input of $\mathscr{A}_i$, that is, their executions are not necessarily independent except their coin tosses.

Definition 4 (Global L_p-sensitivity). For any function $f : \mathscr{D} \to \mathbb{R}^d$, the L_p-sensitivity of f is $\Delta_p f = \max_{D,D'} ||f(D) - f(D')||_p$, for all D, D' differing in at most one record, where $||\cdot||_p$ denotes the L_p-norm.

The Gaussian Mechanism [16] consists of adding Gaussian noise to the true output of a function. In particular, for any function $f : D \to \mathbb{R}^d$, the mechanism is defined as $\mathscr{G}(D) = f(D) + \langle \mathscr{N}_1(0,\sigma), \ldots, \mathscr{N}_d(0,\sigma) \rangle$, where $\mathscr{N}_i(0,\sigma)$ are i.i.d. normal random variables with zero mean and with probability density function $g(z|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}$. The variance σ^2 is calibrated to the L_2-sensitivity of f which is shown by the following theorem.

Theorem 2 ([16]). *For any function $f : \mathscr{D} \to \mathbb{R}^d$, the mechanism $\mathscr{A}$*

$$\mathscr{A}(\mathscr{D}) = f(\mathscr{D}) + \langle \mathscr{G}_1(\sigma), \ldots, \mathscr{G}_d(\sigma) \rangle$$

gives (ε, δ)-differential privacy for any $\varepsilon < 1$ and $\sigma^2 \geq 2(\Delta_2 f)^2 \ln(1.25/\delta)/\varepsilon^2$, where $\mathscr{G}_i(\sigma)$ are i.i.d Gaussian variables with variance σ^2.

For example, if there are d possible locations and f returns the number of visits per location (i.e., the spatial density), then $\Delta_1 f$ equals the maximum number of all visits of any single individual *in any input dataset*, where $\Delta_2 f \leq \Delta_1 f$. If $\Delta_2 f$ is "too" large or ε and/or δ are "too" small, large noise is added providing less accurate visit counts. Also notice that the noise variance is calibrated to the worst-case contribution of any single individual to the output of f, which means that the count of popular locations visited by many individuals can be more accurately released than less popular locations with smaller counts. Indeed, all location counts are perturbed with the same magnitude of noise, hence the signal-to-noise ratio is higher for larger counts providing smaller relative error.

3 Releasing spatio-temporal data

Suppose a geographical region which is composed of a set $\mathbb{L}$ of locations visited by N individuals over a time of interest with T discretized epochs[5]. These locations may represent a partitioning of the region (e.g., all districts of the metropolitan area of a city). The mobility dataset D of N users is a binary data cube with size $N \cdot |\mathbb{L}| \cdot T$, where $D_{i,L,t} = 1$ if individual i visited location L in epoch t otherwise $D_{i,L,t} = 0$. That is, each individual's record (or trajectory) is represented by a binary vector with size $|\mathbb{L}| \times T$. The spatio-temporal density of locations $\mathbb{L}$ is defined by the number of individuals who visited these locations as a function of time. More precisely, there is a time series $\mathbf{X}^L = \langle X_0^L, X_1^L, \ldots, X_{T-1}^L \rangle$ for any location $L \in \mathbb{L}$, where $X_t^L = \sum_{i=1}^N D_{i,L,t}$ and $0 \leq t < T$. $\mathbf{X}^{\mathbb{L}}$ denotes the set of time series of all locations $\mathbb{L}$ and is referred to as the spatio-temporal density of locations $\mathbb{L}$ in the sequel.

[5] An epoch can be any time interval such as a second, a minute, an hour, etc.

In general, any data release is modelled by the execution of data queries. For example, if the querier is interested in the spatio-temporal density of locations $S_L \subseteq \mathbb{L}$ at time $S_T \subseteq \{0,1,\ldots,T-1\}$, then the query $Q(S_L,S_T)$ is computed as $Q(S_L,S_T) = \sum_{L\in S_L, t\in S_T} \sum_{i=1}^{N} D_{i,L,t} = \sum_{L\in S_L, t\in S_T} X_t^L$. This gives rise to at least three approaches for the privacy-preserving release of spatio-temporal density:

Approach 1: compute any query Q on the original data D (or $\mathbf{X}^{\mathbb{L}}$) and release only the anonymized query result $\hat{Q}(S_L,S_T)$;

Approach 2: anonymize the mobility dataset D into $\hat{D}$, then release $\hat{D}$ which can be used to answer any query Q as $\hat{Q}(S_L,S_T) = \sum_{L\in S_L, t\in S_T} \sum_{i=1}^{N} \hat{D}_{i,L,t}$;

Approach 3: compute the density $\mathbf{X}^{\mathbb{L}}$ from the original mobility data D as $X_t^L = \sum_{i=1}^{N} D_{i,L,t}$, and release the anonymized $\hat{\mathbf{X}}^{\mathbb{L}}$, where $\hat{\mathbf{X}}^{\mathbb{L}}$ can be used to answer any query Q.

In Approach 1, a querier can adaptively (i.e., interactively) choose its queries depending on the result of previously answered queries. By contrast, in Approach 2 and 3, the released data are used to answer arbitrary number and type of queries non-interactively (i.e., the queries are independent of each other). In fact, Approach 1, 2 and 3 only differ in their adversary models: Approach 2 and 3 are instantiations of Approach 1 in the non-interactive setting where the possibly adversarial querier must fix all queries before learning any of its results. Specifically, Approach 2 is simply consists of answering $N \cdot |\mathbb{L}| \cdot T$ binary queries at once, where a query returns an element of the cube D. Similarly, in Approach 2, $|\mathbb{L} \cdot T|$ queries can represent the elements of every time series, where all queries are answered together. As detailed in the sequel, the decreased number of queries as well as the non-interactive answering mechanism is the reason that Approach 3 usually outperforms Approach 1 and 2 in practice as long as the only goal is to release $\mathbf{X}^{\mathbb{L}}$ as accurately as possible meanwhile preserving the privacy of individuals. Hence, we will detail a specific solution of Approach 3 in Section 3.3 and briefly review the rest in Section 3.1 and 3.2.

3.1 Approach 1: Anonymization of specific query results

3.1.1 Syntactic anonymization

Privacy breaches may be alleviated by query auditing which requires to maintain all released queries. The database receives a set of counting queries $Q_1(S_{L_1},S_{T_1}),\ldots,Q_n(S_{L_n},S_{T_n})$, and the auditor needs to decide whether the queries can be answered without revealing any single visit or not. Specifically, the goal is to prevent the *full disclosure* of any single visit of any spatio-temporal point in the dataset.

Definition 5 (Full disclosure). $D_{i,L,t}$ is fully disclosed by a query set $\{Q_1(S_{L_1},S_{T_1}),\ldots,Q_n(S_{L_n},S_{T_n})\}$ if $D_{i,L,t}$ can be uniquely determined, i.e., in all pos-

sible data sets D consistent with the answers $\mathbf{c} = (c_1, \ldots, c_n)$ to queries $Q_1, \ldots, Q_n$, $D_{i,L,t}$ is the same.

As each query corresponds to a linear equation on location visits, the auditor can check whether any location visit can be uniquely determined by solving a system of linear equations specified by the queries. To ease notation, let $\mathbf{x} = (x_1, \ldots, x_{N \cdot |\mathbb{L}| \cdot T})$ denote the set of all location visits, i.e., there is a bijection $\alpha : [1,N] \times \mathbb{L} \times [1,T] \rightarrow [1, N \cdot |\mathbb{L}| \cdot T]$ such that $x_{\alpha(i,L,t)} = D_{i,L,t}$. Let $\mathbf{Q}$ be a matrix with n rows and $N \cdot |\mathbb{L}| \cdot T$ columns. Each row in $\mathbf{Q}$ corresponds to a query, which is represented by a binary vector, indexing the visits that are covered by the query. The system of linear equations is described in matrix form as $\mathbf{Qx} = \mathbf{c}$. Hence, the auditor checks whether any x_i can be uniquely determined by solving the following system of equations:

$$\begin{aligned} & \mathbf{Qx} = \mathbf{c} \\ \text{subject to} \quad & x_i \in \{0,1\} \quad \text{for } 1 \leq i \leq N \cdot |\mathbb{L}| \cdot T \end{aligned} \tag{1}$$

In general, this problem is coNP-hard as the variables x_i have boolean values [34]. However, there exists an efficient polynomial time algorithm in the special case when the queries are 1-dimensional, i.e. there is a permutation of $\mathbf{x}$ where each query covers a subsequence of the permutation. Typical examples include range queries. For instance, if locations are ordered according to their coordinates on a space-filling Hilbert curve, then range queries can ask for the total number of visits of locations (over all epochs) that are geographically also close. In the case of 1-dimensional queries, the auditor has to determine the integer solutions of the following system of equations and inequalities:

$$\begin{aligned} & \mathbf{Qx}' = \mathbf{c} \\ \text{subject to} \quad & 0 \leq x'_i \leq 1 \quad \text{for } 1 \leq i \leq N \cdot |\mathbb{L}| \cdot T \end{aligned} \tag{2}$$

Notice that the variables in Eq. (2) are no longer over boolean data and hence Eq. (2) can be solved in polynomial time with any LP solver [55]. The integer solutions of Eq. (2) equals the solutions of Eq. (1) for 1-dimensional location queries [34].

In the general case, when the queries are multi-dimensional, the auditor can also solve Eq. (2), and the final solutions are obtained by rounding: $\hat{x}_i = 1$ if $x'_i > 1/2$ and $\hat{x}_i = 0$ otherwise. In that case, $\hat{\mathbf{x}} \approx \mathbf{x}$ for sufficiently large number of queries [14]. In particular, if each query covers a visit with probability $1/2$, then $O(|\mathbf{x}| \log^2 |\mathbf{x}|)$ queries are sufficient to recover almost the whole $\mathbf{x}$ (i.e., dataset D). Even more, only $|\mathbf{x}|$ number of deterministically chosen queries are enough to recover almost the entire original data [17]. In fact, these reconstruction techniques are the best known attacks against a database curator who answers only aggregate counting queries over boolean data.

Therefore, equipped with the original data $\mathbf{x}$, the auditor can check whether any of the above attacks would be successful by comparing $\mathbf{x}$ with the reconstructed values $\hat{\mathbf{x}}$ (or $\mathbf{x}'$). If so, the auditor refuses to answer any of the n queries.

The above query auditing techniques have several problems. First and foremost, refusing to answer a query itself can leak information about the underlying dataset

(i.e., D) [44]. This would not be the case if refusal was independent of the underlying dataset (e.g., auditing is carried out without accessing the true answers $\mathbf{c}$). Second, they can be computationally expensive. Indeed, using the solver in [55] the worst-case running time is $O(n|\mathbf{x}|^4)$ if $|\mathbf{x}| \gg n$. Finally, most query auditing schemes assume that the adversary has either no background knowledge about the data, or it is known to the auditor. These are impractical assumptions which is also demonstrated in Section 1.1.1, where the adversary reconstructed complete trajectories from aggregate location counts exploiting some inherent characteristics of human mobility.

3.1.2 Semantic anonymization

An alternative approach to query auditing perturbs each query result with some random noise and releases these noisy answers. In order to guarantee (ε, δ)-differential privacy, the added noise usually follows a Laplace or Gaussian distribution. If the noise is added independently to each query answer, then the error is $O(\sqrt{n\log(1/\delta)}/\varepsilon N)$ [16], where N is the number of individuals and n is the number of queries. This follows from the advanced composition property of differential privacy [16]. Therefore, $\tilde{\Omega}(N^2)$ queries can be answered using this approach with non-trivial error (i.e., it is less than the magnitude of the answer). We note that at least $\Omega(\sqrt{N})$ noise is needed per query in order to guarantee any reasonable notion of privacy [14, 16]. There also exist better techniques that add correlated noise to the answers. For instance, the private multiplicative weight mechanism [26] can answer exponentially many queries in N with non-trivial error, where the added noise scales with $O(\sqrt{\log(T|\mathbb{L}|)\cdot\log(1/\delta)}\cdot\log(n)/\varepsilon N)^{1/2}$.

In contrast to query auditing described in Section 3.1, the above mechanisms can answer queries in an on-line fashion (i.e., each query is answered as it arrives) and run in time $poly(N, T|\mathbb{L}|)$ per query. Moreover, the privacy guarantee is independent of the adversarial background knowledge (see Section 2.2). On the other hand, they distort (falsify) the data by perturbation, which may not be desirable in some practical applications of spatio-temporal density. Another drawback is that they are data agnostic and may not exploit some inherent correlation between query results which are due to the nature of the location data. For example, query results usually follow a publicly known periodic trend, and adding noise in the frequency domain can provide more accurate answers [5].

3.2 Approach 2: Anonymization of the mobility dataset

3.2.1 Syntactic anonymization

In general, anonymizing location trajectories (i.e., the whole cube D) while preserving practically acceptable utility is challenging. This is due to the fact that loca-

tion data is typically high-dimensional and sparse, that is, any individual can visit a large number of different locations, but most of them typically visit only a few locations which are quite different per user. This has devastating effect on the utility of anonymized datasets: most k-anonymization schemes generalize multiple trajectories into a single group (or cluster) and represent each trajectory with the centroid of their cluster [2, 43, 47]. Hence, every record becomes (syntactically) indistinguishable from other records within its cluster. This generalization is often implemented by some sophisticated clustering algorithm, where the most similar trajectories are grouped together with an additional (privacy) constraint: each cluster must contain at least k trajectories. Unfortunately, such approaches fail to provide sufficiently useful anonymized datasets because of the *curse of dimensionality* [6]: any trajectory exhibits almost identical similarity to any other trajectory in the dataset. This implies that the centroid of each cluster tend to be very dissimilar from the cluster members implying weak utility. Moreover, as the distribution of the number of visits of spatio-temporal points are typically heavy-tailed [45], projection to low dimensions and then clustering in low dimension also loses almost all information about the trajectories. This is illustrated by Figure 1 which shows the result of a state-of-the-art anonymization scheme, referred to as Never-Walk-Alone (NWA) [2], on a synthetic dataset with 1000 trajectories[6]. This scheme groups k co-localized trajectories within the same time period to form a k-anonymized aggregate trajectory, where k was set to 3 in our experiment and the greatest difference between any spatial point of two members of the same cluster is set to 2000 meters. Figure 1 shows that even with modest values of k, the anonymized dataset provides quite imprecise spatial density of the city.

To improve utility while relaxing privacy requirements, k^m-anonymity has also been considered to anonymize location trajectories in [48]. However, most anonymization solutions guaranteeing k^m-anonymity has a computational cost which is exponential in m in the worst-case, hence this approach is only feasible if m is small. This drawback is alleviated in [3], where a probabilistic relaxation of k^m-anonymity is proposed to release the location visits of individuals without temporal information. In theory, temporal data can also be released along with the location information if the m items are composed of pairs of spatial and temporal positions. However, care must be taken as the background knowledge of a realistic adversary cannot always be represented by m items (e.g., it perhaps also knows the frequency of m items of a targeted individual).

Another approach improving on k-anonymization is p-confidentiality [10]; instead of grouping the trajectories, the underlying map is anonymized, i.e., points of interest are grouped together creating obfuscation areas around sensitive locations. More precisely, given the path of a trajectory, p bounds the probability that the trajectory stops at a sensitive node in any group. Supposing that (i) the background knowledge of the adversary consists of stopping probabilities for each location in a single path and (ii) sensitive locations are pre-specified by data owners, groups of locations are formed in such a way that the parts of trajectories entering the groups

[6] We used a subset of a larger synthetic trajectory dataset available on `https://iapg.jade-hs.de/personen/brinkhoff/generator/`

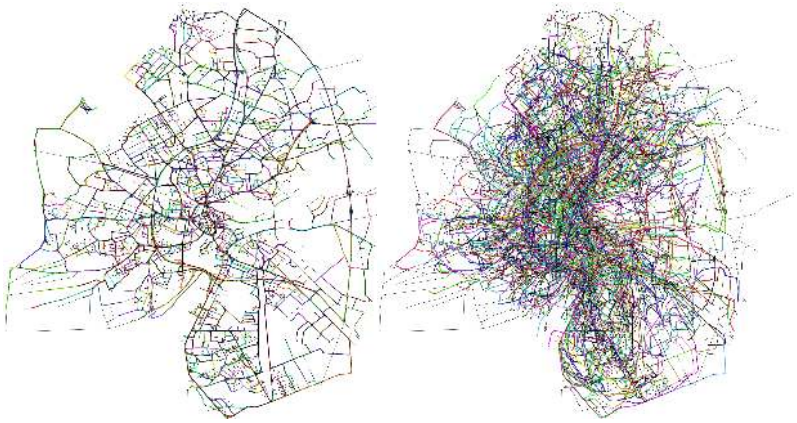


Fig. 1: Never-Walk-Alone anonymization. Original dataset (city of Oldenburg in Germany) with 1000 trajectories (left) and its anonymized version (NWA from [2]) with $k = 3$ where the distance between any points of two trajectories within the same cluster is at most 2000 meters (right). (Image courtesy of Gábor György Gulyás.)

do not increase the adversary's belief in violating the p-confidentiality. Trajectories are then anonymized based on the above map anonymization. The efficiency and utility of this solution is promising, however, in cases where the adversarial background knowledge cannot be approximated well (or at all), semantic privacy models such as differential privacy is preferred.

3.2.2 Semantic anonymization

A more promising approach is to publish a synthetic (anonymized) mobility dataset resembling the original dataset as much as possible, while achieving provable guarantees w.r.t. the privacy of each individual. The records in both datasets follow similar underlying distributions, i.e., after modeling the generator distribution of the original dataset, random samples (records) are drawn from a noisy version of this distribution. A few solutions exist in literature where the generator distribution is modeled explicitly and noised to guarantee differential privacy. For example, DP-WHERE [41] adds noise to the set of empirical probability distributions which is derived from CDR (Call-Detail-Record) datasets, and samples from these distributions to generate synthetic CDRs which are differential private. Although this synthetic dataset can also be used to compute spatio-temporal density, it is usually not as accurate as perturbing the generator distribution of the spatio-temporal density exclusively [4]. Indeed, the accurate model of more complex data (such as the original mobility data) is also more complex in general (i.e., have larger number of parameters), which usually requires increased perturbation.

Some other works generate synthetic sequential data using more general data generating models such as different Markov models [9, 8, 29]. These approaches have wide applicability but they are usually not as accurate as a specific model tailored to the *publicly known* characteristics of the dataset to be anonymized. We illustrate this important point by the following example. DP-WHERE is designed for CDR datasets, and provides more accurate anonymized CDR data than a simple n-gram model [8]. For example, DP-WHERE models the distribution of commute distances per home location and then generates a pair of home and working places as follows. First, a home location is selected, which is followed by picking a distance from the (noisy) distribution of commute distances. Finally, a working place is selected which has this distance from the selected home location. This approach results in more accurate representation of home and working places than using the noisy occurrence counts of different pairs of home and working places like in [8]. This is because commute distances are modeled by an exponential distribution [41], and its single rate parameter can be estimated by the median of the empirical data (i.e., commute distances). Therefore, in DP-WHERE, the probability of a particular pair of home and working location depends on their distance, while in an n-gram model, it depends on the occurrence count of this pair in the original dataset. For instance, New York, as a home location, occurs equally likely with LA and Philadelphia, as working places, in an n-gram model, if these pairs have the same frequency in the original dataset. By contrast, in DP-WHERE, New York is much more likely to co-occur with the geographically closer Philadelphia than with LA. The moral of the story is that achieving the best performance requires to find the most faithful model of the data whose accuracy does not degrade significantly due to additional perturbation.

3.3 Approach 3: Anonymization of spatio-temporal density

A simple k-anonymization of time series $\mathbf{X}^{\mathbb{L}}$ releases X_t^L only if $X_t^L \geq k$. However, as it is detailed in Section 1, this still allows privacy violations through various reconstruction attacks. Hence, releasing spatio-temporal density with provable privacy guarantees, such as differential privacy, is preferred in many practical scenarios.

Within the literature of differential privacy, a plethora of techniques have been proposed to release 1- and 2-dimensional range queries (or histograms) while preserving differential privacy [59, 28, 49, 35, 38, 60, 12, 36, 5, 64, 62, 26] and they are also systematically compared in [27]. Indeed, interpreting query results (or bin counts in a histogram) as location counts, these techniques are directly applicable to release spatial density without temporal data. In theory, low-dimensional embedding, such as Locality-sensitive hashing (LSH) [50], may allow to use any of the above techniques to release spatio-temporal density.

Another line of research addresses the release of time series data with the guarantees of differential privacy. This is challenging as time series are large dimensional data whose global sensitivity is usually so large that the magnitude of the added

noise is greater than the actual counts of the series for stringent privacy requirement (i.e., $\varepsilon < 1$ and $\delta \leq 1/|N|$ where N is the number of records). Consequently, naively adding noise to each count of a time series often results in useless data. Several more sophisticated techniques [51, 31, 19] have been proposed to release time series data meanwhile guaranteeing differential privacy. Most of these methods reduce the global sensitivity of the time series by using standard lossy compression techniques borrowed from signal processing such as sampling, low-pass filtering, Kalman filtering, and smoothing via averaging. The main idea that the utility degradation is decomposed into a reconstruction error, which is due to lossy compression, and a perturbation error, which is due to the injected Laplace or Gaussian noise to guarantee differential privacy. Although strongly compressed data is less accurate, it also requires less noise to be added to guarantee privacy. The goal is to find a good balance between compression and perturbation to minimize the total error.

There are only a few existing papers addressing the release of spatio-temporal density specifically. Although data sources (and hence the definition of spatio-temporal density) vary to a degree in these papers, the commonality is the usage of domain-specific knowledge, i.e., the correlation of data points at hand in both the spatial and the temporal dimension. This domain-specific knowledge helps overcome several challenges including high perturbation error, data sparsity in the spatial domain, and (in some of the cases) real-time data publication. In the context of releasing multi-location traffic aggregates, road network and density are utilized to model the auto-correlation of individual regions over time as well as correlation between neighboring regions [20]. Temporal estimation establishes an internal time series model for each individual cell and performs posterior estimation to improve the utility of the shared traffic aggregate per time stamp. Spatial estimation builds a spatial indexing structure to group similar cells together reducing the impact of data sparsity. All computations are lightweight enabling real-time data publishing. Drawing on the notion of w-event privacy [32], RescueDP studies the problem of the real-time release of population statistics per regions [57]. Such w-event privacy protects each user's mobility trace over any successive w time stamp inside the infinite data grouping algorithm that dynamically aggregates sparse regions together. The criterion for regions to be grouped is that local population statistics should follow a similar trend. Finally, a practical scheme for releasing the spatio-temporal density of a large municipality based on a large CDR dataset is introduced in [4]. Owing to the complexity of its scenario and the innovative techniques used, we present this work in detail in Section 4.

4 A Case-study: Anonymizing the spatio-temporal density of Paris

In this section, we present an anonymization (or sanitization) technique in order to release the spatio-temporal density with provable privacy guarantees. Several optimizations are applied to boost accuracy: time series are compressed by sampling,

clustering and low-pass filtering. The distortion of the perturbation is attenuated via further optimization and post-processing algorithms. A striking demonstration shows that the achieved performance is high and can be practical in real-world applications: the spatio-temporal density of the city of Paris in France, covering roughly 2 million people over 105 km^2, is anonymized using the proposed approach.

The specific goal is to release the spatio-temporal density of 989 non-overlapping areas in Paris, called IRIS cells. Each cell is defined by INSEE[7] and covers about 2000 inhabitants. $\mathbb{L}$ denotes the set of all IRIS cells henceforth, and are depicted in Figure 2 based on their contours[8]. We aim to release the number of all individuals who visited a specific IRIS cell in each hour over a whole week. Since human mobility trajectories exhibit a high degree of temporal and spatial regularity [24], one week long period should be sufficient for most practical applications. Therefore, we are interested in the time series $\mathbf{X}^L = \langle X_0^L, X_1^L, \ldots, X_{167}^L \rangle$ of any IRIS cell $L \in \mathbb{L}$, where X_t^L denotes the number of individuals at L in the $(t+1)$th hour of the week, such that any single individual can visit a tower only once in an hour. We will omit t and L in the sequel, if they are unambiguous in the given context. $\mathbf{X}^{\mathbb{L}}$ denotes the set of time series of all IRIS cells in the sequel.

To compute $\mathbf{X}^{\mathbb{L}}$, we use a CDR (Call Detail Record) dataset provided by a large telecom company. This CDR data contains the list of events of each subscriber (user) of the operator, where an event is composed of the location (GPS coordinate of the cell tower), along with a timestamp, where an incoming/outgoing call or message is sent to/from the individual. The dataset contains the events of $N = 1,992,846$ users at 1303 towers within the administrative region of Paris (i.e., the union of all IRIS cells) over a single week (10/09/2007 - 17/09/2007). Within this interval, the average number of events per user is 13.55 with a standard deviation of 18.33 (assuming that an individual can visit any tower cell only once in an hour) and with a maximum at 732. Similarly to IRIS cells, we can create another set of time series $\mathbf{X}^{\mathbb{C}}$, where X_t^C denotes the number of visits of tower C in the $(t+1)$th hour of the week.

To map the counts in $\mathbf{X}^{\mathbb{C}}$ to $\mathbf{X}^{\mathbb{L}}$, we compute the Voronoi tessellation of the towers cells $\mathbb{C}$ which is shown in Figure 2. Then, we calculate the count of each IRIS cell in each hour from the counts of its overlapping tower cells; each tower cell contributes with a count which is proportional to the size of the overlapping area. More specifically, if an IRIS cell L overlaps with tower cells $\{C_1, C_2, \ldots, C_c\}$, then

$$X_t^L = \sum_{i=1}^{c} X_t^{C_i} \times \frac{\mathsf{size}(C_i \cap L)}{\mathsf{size}(C_i)} \tag{3}$$

at time t.

The rationale behind this mapping is that users are usually registered at the geographically closest tower at any time. Notice that this mapping technique might

[7] National Institute of Statistics and Economics: `http://www.insee.fr/fr/methodes/default.asp?page=zonages/iris.htm`

[8] Available on IGN's website (National Geographic Institute): `http://professionnels.ign.fr/contoursiris`

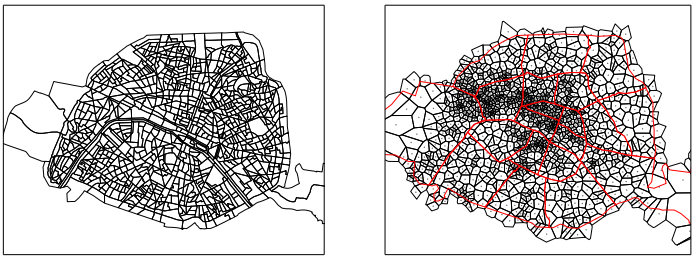


Fig. 2: IRIS cells of Paris (left) and Voronoi-tesselation of tower cells (right)

Algorithm 1 Anonymization scheme

Input: $\mathbf{X}^{\mathbb{T}}$ - input time series (from CDR), (ε, δ)-privacy parameters, $\mathbb{L}$ - IRIS cells, ℓ - maximum visits per user
Output: Noisy time series $\hat{\mathbf{X}}^{\mathbb{L}}$

1: Create $\overline{\mathbf{X}}^{\mathbb{C}}$ by sampling at most ℓ visits per user from $\mathbf{X}^{\mathbb{C}}$
2: Compute the IRIS time series $\overline{\mathbf{X}}^{\mathbb{L}}$ from $\overline{\mathbf{X}}^{\mathbb{C}}$ using Eq, (3)
3: Perturb $\overline{\mathbf{X}}^{\mathbb{L}}$ into $\hat{\mathbf{X}}^{\mathbb{L}}$ `//see Algorithm 2`
4: Apply smoothing on $\hat{\mathbf{X}}^{\mathbb{L}}$

sometimes be incorrect, since the real association of users and towers depends on several other factors such as signal strength or load-balancing. Nevertheless, without more details of the cellular network beyond the towers' GPS position, there is not any better mapping technique.

4.1 Outline of the anonymization process

The aim is to transform the time series of all IRIS cells $\mathbf{X}^{\mathbb{L}}$ to a sanitized version $\hat{\mathbf{X}}^{\mathbb{L}}$ such that $\hat{\mathbf{X}}^{\mathbb{L}}$ satisfies Definition 3. That is, the distribution of $\hat{\mathbf{X}}^{\mathbb{L}}$ will be insensitive (up to ε and δ) to all the visits of any single user during the whole week, meanwhile the error between $\hat{\mathbf{X}}^{\mathbb{L}}$ and $\mathbf{X}^{\mathbb{L}}$ is small.

The anonymization algorithm is sketched in Algorithm 1. First, the input dataset is pre-sampled such that only ℓ visits are retained per user (Line 1). This ensures that the global L_1-sensitivity of all the time series (i.e., $\mathbf{X}^{\mathbb{L}}$) is no more than ℓ. Then, the pre-sampled time series of each IRIS cell is computed from that of the tower cells using Voronoi-tesselation (Line 2), which is followed by the perturbation of the time series of all IRIS cells to guarantee privacy (Line 3). In order to mitigate the distortion of the previous steps, smoothing is applied on the perturbed time series as a post-processing step (Line 4).

4.2 Pre-sampling

To perturb the time series of all IRIS cells, we first compute their sensitivity, i.e., $\Delta_1(\mathbf{X}^{\mathbb{L}})$. To this end, we first need to calculate the sensitivity of the time series of all tower cells, i.e., $\Delta_1(\mathbf{X}^{\mathbb{C}})$. Indeed, Eq. (3) does not change the L_1-sensitivity of tower counts, and hence, $\Delta_1(\mathbf{X}^{\mathbb{C}}) = \Delta_1(\mathbf{X}^{\mathbb{L}})$.

$\Delta_1(\mathbf{X}^{\mathbb{C}})$ is given by the maximum *total* number of (tower) visits of a single user in *any* input dataset. This upper bound must universally hold for all possible input datasets, and is usually on the order of few hundreds; recall that the maximum number of visits per user is 732 in our dataset. This would require excessive noise to be added in the perturbation phase. Instead, each record of any input dataset is truncated by considering at most one visit per hour for each user, and then at most ℓ of such visits are selected per user uniformly at random over the whole week. This implies that a user can contribute with at most ℓ to all the counts in total regardless of the input dataset, and hence, the L_1-sensitivity of the dataset always becomes ℓ. The pre-sampled dataset is denoted by $\overline{\mathbf{X}}$, and $\Delta_1(\overline{\mathbf{X}}^{\mathbb{C}}) = \Delta_1(\overline{\mathbf{X}}^{\mathbb{L}}) = \ell$.

In order to compute the L_2-sensitivity $\Delta_2(\mathbf{X}^{\mathbb{L}})$, observe that, for any t, there is only a single tower whose count can change (by at most 1) by modifying a single user's data. From Eq. (3), it follows that the total change of all IRIS cell counts is at most 1 at any t, and hence $\Delta_2(\overline{\mathbf{X}}^{\mathbb{L}}) \leq \Delta_2(\overline{\mathbf{X}}^{\mathbb{C}}) = \sqrt{\ell}$ based on the definition of L_2-norm.

4.3 Perturbation

The time series $\overline{\mathbf{X}}^{\mathbb{L}}$ can be perturbed by adding $\mathscr{G}(\sqrt{2\ell\ln(1.25/\delta)}/\varepsilon)$ to each count in all time series (see Theorem 2) in order to guarantee (ε,δ)-DP. Unfortunately, this naive method provides very poor results as individual cells have much smaller counts than the magnitude of the injected noise; the standard deviation of the Gaussian noise is 95 with $\varepsilon = 0.3$ and $\delta = 2 \cdot 10^{-6}$, which is comparable to the mean count in $\overline{\mathbf{X}}^{\mathbb{L}}$.

A better approach exploits (1) the similarity of geographically close time series, as well as (2) their periodic nature. In particular, nearby less populated cells are first clustered until their aggregated counts become sufficiently large to resist noise. The key observation is that the time series of close cells follow very similar trends, but their counts usually have different magnitudes. Hence, if we simply aggregate (i.e., sum up) all time series within such a cluster, the aggregated series will have a trend close to its individual components yet large enough counts to tolerate perturbation. To this end, the time series of individual cells are first accurately approximated by normalizing their aggregated time series (i.e., the aggregated count of each hour is divided with the total number of visits inside the cluster), and then scaled back with the (noisy) total number of visits of individual cells.

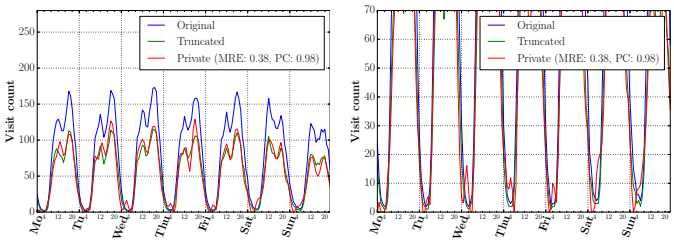
Algorithm 2 Perturbation

Input: Pre-sampled time series $\overline{\mathbf{X}}^{\mathbb{L}}$, Privacy budget ε, δ, Sensitivity $\Delta_1(\overline{\mathbf{X}}^{\mathbb{L}}) = \ell$
Output: Noisy time series $\hat{\mathbf{X}}^{\mathbb{L}}$

1: $\hat{S}_i := \sum_{t=0}^{167} \overline{X}_t^i + \mathscr{G}(2\sqrt{2\ell \ln(2.5/\delta)}/\varepsilon)$ for each $i \in \mathbb{L}$
2: $\mathbb{E} := \text{Cluster}(\mathbb{L}, \hat{S})$
3: **for** each cluster $E \in \mathbb{E}$ **do**
4: $\quad \overline{\mathbf{X}}^E := \langle \sum_{i \in E} \overline{X}_0^i, \sum_{i \in E} \overline{X}_1^i, \ldots, \sum_{i \in E} \overline{X}_{167}^i \rangle$
5: $\quad \hat{\mathbf{X}}^E := \text{FourierPerturb}(\overline{\mathbf{X}}^E, \varepsilon/2, \delta)$
6: $\quad$ **for** each cell $i \in E$ **do**
7: $\quad\quad \hat{\mathbf{X}}^i := \hat{S}_i \cdot (\hat{\mathbf{X}}_t^E / ||\hat{\mathbf{X}}^E||_1)$
8: $\quad$ **end for**
9: **end for**

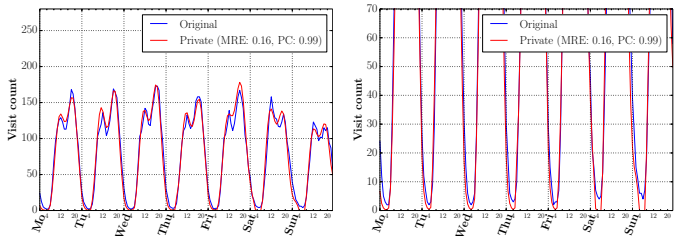
In order to guarantee differential privacy (DP), the aggregated time series are perturbed before normalization. To do so, their periodic nature is exploited and a Fourier-based perturbation scheme [51, 5] is applied: Gaussian noise is added to the Fourier coefficients of the aggregated time series, and all high-frequency components are removed that would be suppressed by the noise. Specifically, the low-frequency components (i.e., largest Fourier coefficients) are retained and perturbed with noise $\mathscr{G}(\sqrt{2\ell \ln(1.25/\delta)}/\varepsilon)$, while the high-frequency components are removed and padded with 0. As only (the noisy) low-frequency components are retained, this method preserves the main trends of the original data more faithfully than simply adding Gaussian noise to $\mathbf{X}^{\mathbb{L}}$, while guaranteeing the same $(\varepsilon/2, \delta/2$-DP. Further details of this technique can be found in [4].

The whole perturbation process is summarized in Algorithm 2. First, the noisy total number of visits of each cell in $\mathbb{L}$ is computed by adding noise $\mathscr{G}(2\sqrt{2\ell \ln(2.5/\delta)}/\varepsilon)$ to $\sum_{t=0}^{167} \overline{X}_t^i$ for cell i (Line 1). These noisy total counts are used to cluster similar cells in Line 2 by invoking any clustering algorithm aiming to create clusters with large aggregated counts overall (i.e., the sum of all cells' time series within the cluster has large counts) using only the noisy total number of visits $\hat{S}_i$ as input. The output $\mathbb{E}$ of this clustering algorithm is a partitioning of cells $\mathbb{L}$. When clusters E are created, their aggregated time series (i.e., the sum of all cells' time series within the cluster) is perturbed with a Fourier-based perturbation scheme [5] in Line 5. Finally, the perturbed time series of each cell i in $\mathbb{L}$ is computed in Line 7 by scaling back the normalized aggregated time series with the noisy total count cell i (i.e., with $\hat{S}_i$). Since Line 1 guarantees $(\varepsilon/2, \delta/2)$-DP to the total counts $(\Delta_1(\overline{\mathbf{X}}^{\mathbb{L}}) = \sqrt{\ell})$, it follows from Theorem 1 that Algorithm 2 is (ε, δ)-DP as the Fourier perturbation of time-series is $(\varepsilon/2, \delta/2)$-DP in Line 5 [4].



(a) Large counts (b) Small counts around local minimas

Fig. 3: Algorithm 1 before improvements ($\varepsilon = 0.3$, $\delta = 2 \cdot 10^{-6}$, $\ell = 30$).



(a) Scaling (b) Smoothing

Fig. 4: Algorithm 1 after improvements ($\varepsilon = 0.3, \delta = 2 \cdot 10^{-6}, \ell = 30$)

4.4 Improvements: Scaling and Smoothing

The result of the above perturbation technique, which is illustrated in Figure 3, still suggests a large error on average. The difference between $\hat{\mathbf{X}}$ and $\mathbf{X}$ is the result of two errors: the sampling error (between $\overline{\mathbf{X}}$ and $\mathbf{X}$) is attributed to pre-sampling, whereas the perturbation error (between $\hat{\mathbf{X}}$ and $\overline{\mathbf{X}}$) is due to the perturbation scheme presented in Algorithm 2. Indeed, since $\overline{\mathbf{X}}^i$ is the pre-sampled time series of cell i, $\hat{\mathbf{X}}^i$ (Line 6 of Algorithm 2) will be a scaled down version of the original time series $\mathbf{X}^i$ due to the fact that the ℓ visits per individual are sampled *uniformly* at random.

As illustrated by Figure 3a, sampling error mainly distorts large counts: although the noisy counts are close to the counts of the *truncated* (pre-sampled) time series between 9:00 AM and 11:00 PM, it is still far from the original count values. This significantly increases the mean relative error. In addition, as Figure 3b also shows, noisy counts also deviate from pre-sampled as well as from original counts around the local minimas (close to 4:00 AM every day), which further deteriorates the relative error.

To alleviate these errors, two further improvements are proposed in [4]: first, the perturbation of total cell counts (Line 1 in Algorithm 2) is improved, which is used in cell clustering (Line 2 in Algorithm 2) and scaling (Line 6 in Algorithm 2). The main idea is that the real scaling factor $\sum_{t=0}^{167} X_t^i$ (in Line 1 of Algorithm 2) is approximated by a more accurate technique: the relative frequency of each tower is first estimated by sampling only a single visit per user, then the perturbed relative frequencies are multiplied with the (perturbed) total number of visits of the original data $\mathbf{X}$ to obtain an estimation of $\sum_{t=0}^{167} X_t^i$. The relative frequencies have L_2-sensitivity 1, while the L_2-sensitivity of the total number of visits is $\sqrt{753} < 27.44$. Hence, the relative error of this new estimation becomes small, as the relative frequencies of towers require very small noise, while the total number of visits is incomparably larger than its L_2-sensitivity. Finally, in order to diminish perturbation error of small counts, counts between 0:00 and 6:00 AM are smoothed out through non-linear least-square fitting as a post-processing step.

4.5 Time complexity

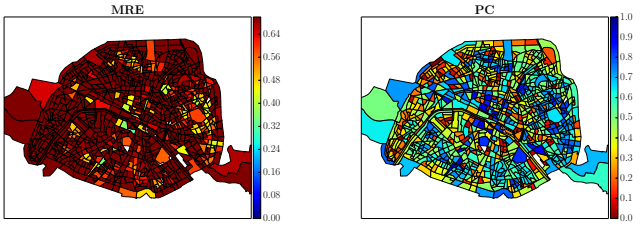
The pre-sampling step has a complexity of $O(\ell N)$ and the computation of $\mathbf{X}^{\mathbb{L}}$ (see Eq. (3)) needs $O(T|\mathbb{C}||\mathbb{L}|)$ steps in the worst case. In the perturbation algorithm (Alg. 2), the clustering of time-series runs in $O(T|\mathbb{L}|^2)$ and the Discrete Cosine Transform can be implemented with Fast Fourier Transform that has a complexity of $O(T \log T)$. Therefore, the overall complexity is $O(|\mathbb{L}|T \log T + T|\mathbb{L}|^2 + T|\mathbb{C}||\mathbb{L}| + \ell N)$ disregarding the post-processing step (in Line 4 of Algorithm 1).

4.6 Results

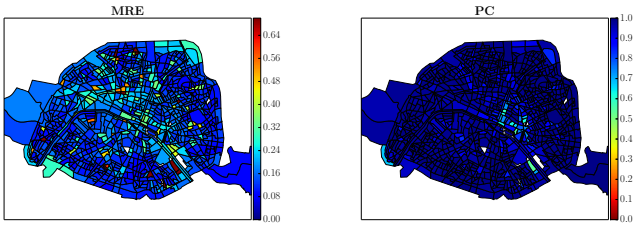
The error between the anonymized and original time series is measured by two metrics: the mean relative error (MRE) and the Pearson Correlation (PC), where $\mathrm{MRE}(\mathbf{X}, \hat{\mathbf{X}}) = (1/n)\sum_{i=0}^{n-1} \frac{|\hat{X}_i - X_i|}{\max(\gamma, X_i)}$[9]. The Pearson correlation measures the linear correlation between the noisy and the original time series (i.e., whether they have similar trends), and it always falls between -1 and 1.

The MRE and PC of individual IRIS cells are illustrated by color maps in Figure 5. This figure shows that the presented anonymization (Figure 5b) scheme outperforms the naive Gaussian Perturbation Algorithm (Figure 5a) when $\mathscr{G}(\sqrt{2\ell \ln(1.25/\delta)}/\varepsilon)$ is added to each count in $\overline{\mathbf{X}}^{\mathbb{L}}$ without any further optimization. Moreover, Algorithm 1 can also provide practical utility for most cells with strong privacy guarantee. Specifically, the average MRE over all cells is only 0.17 with $\varepsilon = 0.3$, $\delta = 2 \cdot 10^{-6}$ and $\ell = 30$.

[9] The sanity bound γ mitigates the effect of very small counts and is adjusted to 0.1% of $\sum_{i=0}^{n-1} X_i$ [58]



(a) Naive Gaussian Perturbation (Avg. MRE: 1.01, PC: 0.47)



(b) Algorithm 1 (Avg. MRE: 0.17, PC: 0.96)

Fig. 5: Mean Relative Error and Pearson Correlation of each IRIS cell ($\varepsilon = 0.3$, $\delta = 2 \cdot 10^{-6}$, $\ell = 30$)

5 Summary and Conclusions

In this chapter, we gave an overview of the privacy models and anonymization/sanitization techniques for releasing spatio-temporal density in a privacy-preserving manner. We first illustrated the privacy threats of releasing spatio-temporal density and described two attacks that can recover individual visits or even complete trajectories merely from spatio-temporal density. Then, we reviewed the mainstream privacy models, and distinguished syntactic models (such as k-anonymity) and semantic models (such as differential privacy). As spatio-temporal density is a function of the raw mobility data, we identified three main approaches to anonymize spatio-temporal density: (1) anonymize and release the results of queries executed on the original mobility data, (2) anonymize and release the original mobility data (i.e., location trajectories) used to compute the spatio-temporal density, and (3) anonymize and release the spatio-temporal density directly which is computed from the original mobility data.

The first approach relies on query auditing, or query perturbation using differential privacy. Query auditing is computationally expensive, and disregards the background knowledge of the adversary. Although query perturbation is independent of the adversarial background knowledge and runs in polynomial time, it ignores some inherent characteristics of human mobility which could further diminish perturbation error. Also, unlike query auditing, perturbation is non-truthful, i.e. releases falsified location data.

The second approach can use either a syntactic or a semantic privacy model to anonymize trajectories. Syntactic anonymization techniques providing k-anonymity suffer from the curse of dimensionality and provide inaccurate data in general. k^m-anonymization has smaller error but guarantees weaker privacy and/or has exponential time complexity in m. In addition, all syntactic privacy guarantees can be violated with appropriate background knowledge, which is difficult to model in practice. Semantic anonymization using differential privacy is much more promising, but again, they use perturbation which is non-truthful. In addition, anonymizing trajectories usually provides less accurate density estimation than anonymizing the spatio-temporal density directly. Indeed, density can be modelled accurately with a model which requires less perturbation than the model of complete trajectories. Although some trajectory anonymization techniques have larger time complexity, these are not serious concerns in case of one-shot release.

As the last approach provides the largest accuracy in practice, we detailed the operation of such an anonymization process and showed its performance in a real-world application. This demonstration also shows that differential privacy can be a practical model for the privacy-preserving release of spatio-temporal data, even if it has large dimension. We also showed that, in order to achieve meaningful accuracy, the sanitization process has to be carefully customized to the application and public characteristics of the dataset. The time complexity of this approach is polynomial and also very fast in practice.

As a conclusion, it is unlikely that there is any "universal" anonymization/sanitization solution that fits every application and data, i.e., provides good accuracy in all scenarios. In particular, achieving the best performance requires finding the most faithful model of the data, such that it withstands perturbation. In case of spatio-temporal density, clustering and sampling with Fourier-based perturbation are seemingly the best choices due to the periodic nature and large sensitivity of location counts.

Finally, we emphasize two important properties of semantic anonymization and query perturbation with differential privacy. First, unlike all other schemes, including query auditing and syntactic trajectory anonymization, differential privacy composes and the privacy loss can be quantified and gracefully degrades by multiple releases. This is crucial if the data gets updated and should be "re-anonymized", or, there are other independent releases with overlapping set of individuals (e.g., two CDR datasets about the same city from two different telecom operators). Second, privacy attacks may rely on very diverse background knowledge, which are difficult to capture. For example, not until the appearance of the reconstruction attack in Section 1.1.1 was it clear that individual trajectories can be recovered merely from spatio-temporal density. Only differential privacy seems to provide adequate defense (with properly adjusted ε and δ) against even such sophisticated attacks.

Nevertheless, there are still many interesting future directions to further improve performance. First, the data generating distribution can be implicitly modeled using generative artificial neural networks (ANNs) such as Recurrent Neural Networks (RNNs) [25]. Generative ANNs have exhibited great progress recently and their representational power has been demonstrated by generating very realistic (but still

artificial) sequential data such as texts[10] or music. The intuition is that, as deep ANNs can "automatically" model very complex data generating distributions thanks to their hierarchical structure, they can potentially be used to produce realistic synthetic sequential data such as spatio-temporal densities. Second, current approaches release the spatio-temporal density only for a limited time interval. For example, the solution described in Section 4 releases the density for only a single week. To release density over multiple weeks, one need to use a the composition property of differential privacy which guarantees $(k\varepsilon, k\delta)$-DP for k-fold adaptive composition based on Theorem 1. These are still quite large bounds if we wish to release the density in the whole year with $k = 52$. Fortunately, tighter bound has been derived recently, building on the notions of Concentrated Differential Privacy, which guarantees $\left(O(\varepsilon\sqrt{k}), \delta\right)$-DP after k adaptive releases [1].

Acknowledgments

Gergely Acs has been supported by the MTA Premium Post-doctoral Fellowship of the Hungarian Academy of Sciences. Gergely Biczók has been supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *ACM CCS*, 2016.
2. O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.
3. G. Acs, J. P. Achara, and C. Castelluccia. Probabilistic k^m-anonymity (Efficient Anonymization of Large Set-Valued Datasets). In *IEEE International Conference on Big Data (Big Data)*, 2015.
4. G. Acs and C. Castelluccia. A Case Study: Privacy Preserving Release of Spatio-temporal Density in Paris. In *KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2014.
5. G. Acs, R. Chen, and C. Castelluccia. Differentially private histogram publishing through lossy compression. In *ICDM*, 2012.
6. C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, 2005.
7. L. Bengtsson, X. Lu, A. Thorson, R. Garfield, and J. Von Schreeb. Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in haiti. *PLoS Med*, 8(8):e1001083, 2011.
8. R. Chen, G. Acs, and C. Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM Conference on Computer and Communications Security*, pages 638–649, 2012.
9. R. Chen, B. C. M. Fung, and B. C. Desai. Differentially private trajectory data publication. *CoRR*, abs/1112.2020, 2011.

[10] `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`

10. A. E. Cicek, M. E. Nergiz, and Y. Saygin. Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *The VLDB Journal*, 23(4):609–625, 2014.
11. G. Cormode. Personal privacy vs population privacy: learning to attack anonymization. In *KDD*, pages 1253–1261, 2011.
12. G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *ICDE*, pages 20–31, 2012.
13. Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports, Nature*, March 2013.
14. I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.
15. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
16. C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4), 2014.
17. C. Dwork and S. Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *CRYPTO*, 2008.
18. European Commission. General European Data Protection Regulation (GDPR). `http://www.privacy-regulation.eu/en/index.htm`, 2016.
19. L. Fan and L. Xiong. Real-time aggregate monitoring with differential privacy. In *ACM CIKM*, pages 2169–2173, 2012.
20. L. Fan, L. Xiong, and V. 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.
21. B. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR)*, 42(4):14, 2010.
22. S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, pages 265–273, 2008.
23. P. Golle. Revisiting the uniqueness of simple demographics in the US population. In *ACM WPES*, pages 77–80, 2006.
24. M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453, 2008.
25. I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT Press, 2016.
26. M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In *NIPS*, 2012.
27. M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang. Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD' 16, pages 139–154, 2016.
28. M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *PVLDB*, 2010.
29. X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava. DPT: differentially private trajectory synthesis using hierarchical reference systems. *PVLDB*, 8(11):1154–1165, 2015.
30. T. Imielinski and W. L. Jr. On th undecidability of equivalence problems for relational expressions. In *Advances in Data Base Theory, Vol. 2, Based on the Proceedings of the Workshop on Logical Data Bases*, pages 393–409, 1982.
31. G. Kellaris and S. Papadopoulos. Practical differential privacy via grouping and smoothing. In *VLDB*, pages 301–312, 2013.
32. G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment*, 7(12):1155–1166, 2014.
33. R. Kitchin. The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014.
34. J. M. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *ACM PODS*, pages 86–91, 2000.
35. C. Li, M. Hay, G. Miklau, and Y. Wang. A data- and workload-aware algorithm for range queries under differential privacy. *Proc. VLDB Endow.*, 7(5):341–352, Jan. 2014.

36. C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, pages 123–134, 2010.
37. N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.
38. N. Li, W. Yang, and W. Qardaji. Differentially private grids for geospatial data. In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE'13, pages 757–768. IEEE Computer Society, 2013.
39. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.
40. F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30, 2009.
41. D. J. Mir, S. Isaacman, R. Cáceres, M. Martonosi, and R. N. Wright. Dp-where: Differentially private modeling of human mobility. In *BigData Conference*, pages 580–588, 2013.
42. N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. Lee. Anonymizing healthcare data: a case study on the blood transfusion service. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 1285–1294, 2009.
43. A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.
44. S. U. Nabar, K. Kenthapadi, N. Mishra, and R. Motwani. A survey of query auditing techniques for data privacy. In *Privacy-Preserving Data Mining - Models and Algorithms*, pages 415–431. 2008.
45. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy (S&P)*, pages 111–125, 2008.
46. P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano. Current trends in smart city initiatives: Some stylised facts. *Cities*, 38:25–36, 2014.
47. M. E. Nergiz, M. Atzori, Y. Saygin, and B. Güç. Towards trajectory anonymization: a generalization-based approach. *Trans. Data Privacy*, 2(1):47–75, 2009.
48. G. Poulis, S. Skiadopoulos, G. Loukides, and A. Gkoulalas-Divanis. Distance-based k^m-anonymization of trajectory data. *IEEE MDM*, 2:57–62, 2013.
49. W. Qardaji, W. Yang, and N. Li. Understanding hierarchical methods for differentially private histograms. *Proc. VLDB Endow.*, 6(14):1954–1965, Sept. 2013.
50. A. Rajaraman and J. Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.
51. V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, 2010.
52. L. Sun, D.-H. Lee, A. Erath, and X. Huang. Using smart card data to extract passenger's spatio-temporal density and train's trajectory of mrt system. In *Proceedings of the ACM SIGKDD international workshop on urban computing*, pages 142–148. ACM, 2012.
53. L. Sweeney. *k*-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.
54. M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. *VLDB Endow.*, 1(1), 2008.
55. P. M. Vaidya. An algorithm for linear programming which requires o(((m+n)nˆ2 + (m+n)ˆ1.5 n)l) arithmetic operations. In *ACM STOC*, pages 29–38, 1987.
56. N. Victor, D. Lopez, and J. H. Abawajy. Privacy models for big data: a survey. *International Journal of Big Data Intelligence*, 3(1):61–75, 2016.
57. Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. 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.
58. X. Xiao, G. Bender, M. Hay, and J. Gehrke. iReduct: Differential privacy with reduced relative errors. In *SIGMOD*, pages 229–240, 2011.
59. X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. In *ICDE*, pages 225–236, 2010.

60. Y. Xiao, L. Xiong, L. Fan, S. Goryczka, and H. Li. Dpcube: Differentially private histogram release through multidimensional partitioning. *Trans. Data Privacy*, 7(3):195–222, Dec. 2014.
61. F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin. Trajectory recovery from ash: User privacy is NOT preserved in aggregated mobility data. In *WWW 2017*, pages 1241–1250, 2017.
62. J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 32–43, 2012.
63. H. Zang and J. Bolot. Anonymization of location data does not work: a large-scale measurement study. In *MOBICOM*, pages 145–156, 2011.
64. X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 587–595, 2014.